

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA: INGENIERÍA DE SISTEMAS

**Trabajo de titulación previo a la obtención del título de: INGENIERO DE
SISTEMAS**

TEMA:

**ANÁLISIS, DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN
SISTEMA INFORMÁTICO PARA GESTIONAR Y CONTROLAR LOS
RESULTADOS DE ANÁLISIS DE LECHE REALIZADO POR EL
LABORATORIO DE CALIDAD DE LECHE DEL CENTRO DE APOYO
CAYAMBE, DE LA UNIVERSIDAD POLITÉCNICA SALESIANA**

AUTOR:

ORLANDO VICENTE HONORES LEÓN

DIRECTORA:

LINA PATRICIA ZAPATA MOLINA

Quito, julio de 2015

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO
DEL TRABAJO DE TITULACIÓN**

Yo, autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaro que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del autor.

Quito, julio de 2015

Orlando Vicente Honores León

C.C. 171478425-1

DEDICATORIA

A Dios, por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

Mi madre Mariana León, por darme la vida, quererme mucho, creer en mí y porque siempre me apoyaste. Mami gracias por darme una carrera para mi futuro, todo esto te lo debo a ti.

Mi padre Adriano Honores, por darme la vida, quererme mucho, por ser estar junto a mi mami y darle el soporte que ella siempre necesitó.

Mis hermanos, Edwin, Sarita y Diego por estar conmigo y apoyarme siempre, los quiero mucho.

Mi tutora de tesis Ing. Lina Zapata, por estar junto a mí durante el transcurso del proyecto de mi tesis, por brindarme sus conocimientos y gran parte de su tiempo

AGRADECIMIENTO

A la Universidad Politécnica Salesiana por aportar en el desarrollo de esta tesis

RESUMEN

Se presenta un sistema realizado en java en conjunto con nuevas tecnologías como NodeJS, AngularJs y Socket IO, que permite la interpretación de los resultados en forma automatizada de los analizadores de leche con los que cuenta el Laboratorio de Cayambe de la Universidad Salesiana.

El uso de este conjunto de herramientas permite la gestión en tiempo real de los requerimientos del laboratorio, como el manejo de datos personales de usuarios y clientes que son validados en línea con las páginas expuestas por el gobierno como el SRI y CNE.

Además brinda una administración del sistema informático en forma dinámica, es decir si surge algún cambio este sería implementado mediante la página de configuración del sistema, la cual permite crear pantallas según la tabla de la base de datos creada.

ABSTRACT

A system made in java along with new technologies as NodeJS, AngularJs and socket IO, which allows the interpretation of the results in automated analyzers milk are there Cayambe Laboratory of the Salesian University is presented.

Using this set of tools allow real-time management of laboratory requirements , such as the handling of personal data of users and customers are validated in line with the pages exposed by the government as the SRI and CNE.

In addition a good administration computer system dynamically, by example if a change this would be implemented by the system configuration page appears, which lets you create screens per table of database created.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1.....	2
PRESENTACIÓN Y FORMULACIÓN DEL PROBLEMA.....	2
1.1 Planteamiento del problema	2
1.2 Definición del problema.....	2
1.3 Formulación de objetivos	3
1.4 Justificación del problema.....	4
1.5 Alcance del problema.....	5
CAPÍTULO 2.....	7
MARCO TEÓRICO	7
2.1 Métodos instrumentales para el análisis de leche cruda	7
2.1.1 Bactoscan	7
2.1.2 Fossomatic.....	7
2.1.3 Milkoscan.....	7
2.1.4 Datos obtenidos de los equipos analizadores	7
2.2 Metodología	8
2.2.1 OOHDM.....	8
2.2.1 1 Fases de la metodología OOHDM	8
2.2.2 UML	13
2.3 Herramientas de desarrollo.....	17
2.4 Pool de conexiones con la base de datos.....	24
2.5 Eclipse como Ide, Entorno de desarrollo integrado.....	25
2.6 Servidores	25
2.6.1 Node.js.....	25
2.6.2 Apache Tomcat.....	25
2.6.3 Postgresql.....	26

2.7 Reportes	28
2.7.1 JasperReports.....	28
2.7.2 IReports.....	28
2.8 Sistema operativo	29
2.8.1 Ubuntu.....	29
2.8.2 Microsoft Windows.....	29
CAPÍTULO 3.....	30
ANÁLISIS Y DISEÑO DE PROCESOS	30
3. 1 Análisis de la situación actual	30
3.1.1 Levantamiento de procesos	31
3.1.2 Análisis de requerimientos del sistema	34
3.2 Diseño	38
3.2.1 Requerimientos funcionales	38
3.2.1.1 Identificación de roles y funcionalidades.	45
3.2.1.2 Especificación de escenarios.....	50
3.2.1.3 Especificación de diagramas de interacción de usuario	54
3.2.2 Diseño conceptual.....	56
3.2.2.1 Especificación y diagramas de casos de uso.....	57
3.2.3 Diseño navegacional	71
3.2.3.1 Esquema de contexto navegacional	71
3.2.4 Diseño de interfaz abstracta.....	72
3.2.4.1 Diagrama relacional.....	76
CAPÍTULO 4.....	78
DESARROLLO DEL SISTEMA PROPUESTO	78
4.1 Módulos de la aplicación	78
4.2 Creación de la aplicación usando Maven.....	81
4.3 Estándares de programación	88

4.4 Codificación del sistema	93
CAPÍTULO 5.....	96
IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA	96
5.1 Implementación del sistema informático	96
5.2 Pruebas.....	97
5.2.1 Pruebas funcionales.	97
5.2.2 Pruebas de validación.	102
CONCLUSIONES.....	104
RECOMENDACIONES.....	106
LISTA DE REFERENCIAS	107
ANEXOS.....	112

ÍNDICE DE FIGURAS

Figura 1. Se muestran las fases de OOHMD.	9
Figura 2. Las cinco fases de análisis de requerimientos.	10
Figura 3. Ejemplo de diseño conceptual.	11
Figura 4. Ejemplo del diseño de interfaz abstracta.	12
Figura 5. Ejemplo de caso de uso para un modelado simple de restaurante.	14
Figura 6. Ejemplo de componentes de un diagrama de casos de uso.	14
Figura 7. Ejemplo de diagrama de clases.	15
Figura 8. Ejemplo de Diagrama de secuencia.	16
Figura 9. Ejemplo de Diagrama E-R.	17
Figura 10. Ciclo de vida de JSF.	21
Figura 11. Representación de AngularJs como front-end.	23
Figura 12. Ejemplo de creación de un módulo en AngularJs.	23
Figura 13. Estructura usando Pool de conexiones.	25
Figura 14. Funcionalidades del sistema.	34
Figura 15. Administración varios tipos.	55
Figura 16. UID Administración de usuarios.	55
Figura 17. UID Gestión de clientes.	55
Figura 18. Diagrama de clase del sistema de software.	56
Figura 19. Caso de Uso Administrar varios tipos.	58
Figura 20. Diagrama administrar menús.	59
Figura 21. Diagrama de caso de uso administrar archivos.	62
Figura 22. Diagrama caso de uso requerimientos.	70
Figura 23. Diagrama caso de uso de recibimiento de muestras de leche para el análisis.	70
Figura 24. Diseño perfil laboratorista.	71
Figura 25. Diseño navegacional.	72
Figura 26. Plantilla de la página principal.	74
Figura 27. Adv. Listar registrar, para ver los registros del sistema.	74
Figura 28. Adv. Edición de datos, para la modificación de registros.	75
Figura 29. Diagrama modelo relacional de la base de datos en Postgres.	76
Figura 30. Diagrama de secuencia.	77
Figura 31. Módulos de la aplicación.	78

Figura 32. Diagrama de clases de clientes.....	79
Figura 33. Diagrama de clases requerimientos.	79
Figura 34. Diagrama de clases recepción de muestras.	80
Figura 35. Diagrama de clases análisis de muestras y entrega de resultados.	80
Figura 36. Estructura del proyecto en Eclipse.	81
Figura 37. Módulos del proyecto.	82
Figura 38. Módulos en maven.	83
Figura 39. Muestra cómo se crea las entidades a partir de la herramienta JPA tolos.	84
Figura 40 Estructura del servidor NodeJs.	86
Figura 41. Pantalla de inicio del sistema.....	99
Figura 42. Configuración de pantallas dinámicas.	100
Figura 43. Pantalla ingreso de datos.....	100
Figura 44. Pantalla estándar para visualizar los registros.	101
Figura 45. Pantalla de informe.	101
Figura 46. Pantalla de administración de perfiles.....	102
Figura 47. Clases que conforman el paquete “validaciones”.	102
Figura 48. Pantalla de validación de la cédula.	103
Figura 49. Pantalla ingreso de email y validación del mismo.	103

ÍNDICE DE TABLAS

Tabla 1. Tipos de diagramas en UML.....	13
Tabla 2. Anotaciones en la persistencia de campos	19
Tabla 3. Rol administrador.....	46
Tabla 4. Rol Jefe de laboratorio	47
Tabla 5. Rol recepcionista.....	48
Tabla 6. Rol laboratorista e invitado	49
Tabla 7. Casos de uso administrar varios tipos.	57
Tabla 8. Caso de uso administrar menús.....	58
Tabla 9. Casos de uso administrar usuarios.	60
Tabla 10. Casos de uso administrar la configuración de archivos aceptados por el sistema.....	61
Tabla 11. Casos de uso administrar perfiles.....	62
Tabla 12. Caso de uso gestionar clientes	63
Tabla 13. Caso de uso gestionar requerimientos que ofrece el laboratorio	64
Tabla 14. Casos de uso recibimiento de muestras de leche para el análisis.....	66
Tabla 15. Casos de uso gestionar reportes.	67
Tabla 16. Casos de uso gestionar normas del Inen.	68
Tabla 17. Caso de uso para gestionar la configuración de pantallas que tiene el sistema.....	69
Tabla 18. Caso de prueba: inicio de sesión	97
Tabla 19. Caso de prueba: administrar usuarios	97
Tabla 20. Gestionar acciones de perfil	98

ÍNDICE DE ANEXO

Anexo 1. Manual de usuarios.....	112
----------------------------------	-----

INTRODUCCIÓN

El presente trabajo se desarrolla en un ambiente informático e investigativo, utilizando herramientas que se encuentran en boga en este momento como lo es JsF, AngularJs, NodeJS combinadas con el lenguaje de java, que permitieron que se desarrolle un sistema en ambiente web para la lectura de archivos de texto que arrojan los equipos analizadores de leche del Laboratorio de calidad de leche de la Universidad Politécnica Salesiana, además permite la visualización de estos archivos a través de reportes y gráficas sobre la calidad de leche a cada cliente.

El desarrollo del tema se lo detalla a continuación: el primero capítulo trata el planteamiento, definición, justificación y alcance del problema del Laboratorio de calidad de leche de la Universidad Politécnica Salesiana, así como la formulación de objetivos. El segundo capítulo se describe el marco teórico y las herramientas a utilizar para la creación del sistema informático. En el tercer capítulo describe el proceso del sistema informático mediante la metodología utilizada OOHDM, en el cual se documenta cada uno de los pasos de este método.

El cuarto capítulo manifiesta el desarrollo del sistema informático para el análisis de calidad de leche del Laboratorio de calidad de leche de la Universidad Politécnica Salesiana. El quinto capítulo se detalla la implantación y pruebas realizadas en el sistema informático. Finalmente se añaden las conclusiones y recomendaciones de este trabajo.

CAPÍTULO 1

PRESENTACIÓN Y FORMULACIÓN DEL PROBLEMA

1.1 Planteamiento del problema

El Laboratorio de calidad de leche de la Universidad Politécnica Salesiana, cuenta desde el año 2008 con el mejor equipamiento tecnológico e innovador para el análisis de calidad de muestras de leche, lo que ha permitido que el laboratorio proporcione un servicio profesional, flexible, ágil y confidencial de análisis de la calidad de la leche a los ganaderos e industrias de lácteos de todo el país en especial de sectores aledaños. Por otra parte el laboratorio apoya en el desarrollo de proyectos técnicos de investigación tanto a nivel local como a nivel nacional y adicionalmente el laboratorio dispone de un sistema informático denominado Qualion, de procedencia brasileña, como complemento al equipamiento tecnológico del laboratorio, pero este software tiene limitadas funciones por cuestiones de licencia y no reúne los requerimientos funcionales reales del laboratorio ya el mismo fue construido para cubrir requerimientos de los laboratorios de la Rede Brasileira de Laboratorios de Controle de Qualidade do Leite – RBQL.

Actualmente el Laboratorio de Calidad de leche es considerada como una unidad de apoyo a las investigaciones de la calidad de leche, en estos últimos años se ha incrementado notablemente la demanda de servicios de análisis de calidad de muestras de leche, y ante la falta de un software que ayude a la elaboración y análisis de resultados finales, ha ocasionado que estos procesos sean lentos, tediosos y se retrase la entrega de resultados finales.

En resumen el laboratorio tiene la necesidad prioritaria de disponer de un sistema informático que se ajuste a sus necesidades, en el procesamiento de datos referentes al análisis de muestras de leche, a fin de que los análisis y entrega de resultados se elaboren con mayor eficiencia y en forma completa.

1.2 Definición del problema

Actualmente el Laboratorio de Calidad de Leche, lleva un estimado de 20 clientes activos, que demandan el servicio de análisis de calidad de muestras de leche; con un promedio de 90 solicitudes diarias, debido a que el sistema informático actual es

muy limitado porque no se ajusta a las necesidades reales propias del Laboratorio, ha ocasionado que algunos procesos importantes como la emisión de resultados finales sean lentos y tediosos de elaborar. Por otra parte la información referente al análisis de muestras de leche, que se migra a Excel, no cumple el formato adecuado para su procesamiento y para mejorarlo se pasa por el proceso de edición manual para obtener información útil que genere los resultados esperados.

Otro proceso que tiene inconveniente, dentro del sistema actual, es la falta de control sobre el material entregado, para el almacenamiento de las muestras de leche a analizarse, a los usuarios que acuden al Laboratorio.

Cabe mencionar que el sistema actual no se ajusta a las necesidades reales del Laboratorio si no por el contrario el Laboratorio ha tenido que ajustarse a la funcionalidad del sistema, quitando eficiencia y rapidez en el procesamiento de datos y por ende en la generación de resultados finales.

1.3 Formulación de objetivos

Objetivo general

Desarrollar un sistema informático, para la web, que permita gestionar y controlar los análisis de las diferentes muestras de leche que ingresan al Laboratorio de Calidad de Leche del Centro de Apoyo, Cayambe.

Objetivos específicos

- Llevar un control automatizado del ingreso de usuario al manejo de información relacionado al análisis de muestras de la leche. Estableciendo para ello diferentes niveles de acceso y el uso de un nombre de usuario y contraseña (único) por usuario.
- Contar con un registro permanente y actualizado sobre los datos personales de cada cliente/empresa que utiliza el Laboratorio de Calidad de Leche para el análisis de muestra de leche.
- Realizar un seguimiento automatizado de los materiales entregados, devueltos o pendientes en su entrega, para cada uno de los clientes que soliciten el servicio de análisis de leche.

- Llevar un registro permanente y actualizado, sobre el estado en que se encuentran las muestras de leche al momento en que son recibidas en el laboratorio, para su posterior análisis.
- Registrar y procesar, en forma automatizada, los resultados obtenidos de las muestras de leche, una vez que pasa el proceso de análisis en el Laboratorio de Calidad de Leche.
- Generar los respectivos reportes y/o consultas automatizadas, referentes a los resultados obtenidos en el análisis de leche.
- Realizar gráficos estadísticos, en forma automatizada, sobre los resultados que se obtiene en cada muestra de leche analizada, de acuerdo al tipo de cliente y considerando los históricos por cliente.

1.4 Justificación del problema

Ante los inconvenientes que tiene el laboratorio tanto en el procesamiento de los resultados obtenidos del análisis de calidad de las muestras de leche como en la generación de informes de resultados con información válida, clara y precisa para el cliente, el Laboratorio tiene la necesidad prioritaria de construir e implementar un nuevo sistema informático que se ajuste a las necesidades reales y propias del Laboratorio.

Con el nuevo sistema informático se busca llevar un mejor control de clientes, permitiendo la categorización de los mismos, debido a que la información que se emite al final debe estar en función del tipo del cliente/empresa y a sus requerimientos y necesidades. Además, con el nuevo sistema se busca mejorar el control sobre la cantidad de muestras de leche recibidas y sobre ciertos factores que afecta directamente a los resultados de la calidad de leche, como temperatura, fecha de colecta, y materiales usados, información que es registrada al momento de la entrega de muestras.

Como parte esencial, con el nuevo sistema informático, se quiere mejorar radicalmente la generación de reportes y estadísticas, debido a que son actualmente procesos manuales que demanda tiempo y esfuerzo por parte del personal del Laboratorio. Con esto los clientes externos o internos, obtendrán sus resultados finales anexados a los gráficos estadísticos de forma más rápida y oportuna,

información útil y dinámica como herramienta para la generación de proyectos de investigación, capacitación o control a nivel de país.

1.5 Alcance del problema

Para identificar los requerimientos funcionales del sistema informático propuesto, se interactuó directamente con los usuarios involucrados mediante entrevistas abiertas o cerradas, se realizaron visitas constantes al Laboratorio de Calidad de Leche para la revisión de formularios físicos, que contienen información referente a las muestras de leche y la descripción de los resultados obtenidos en su análisis de calidad.

El presente proyecto de tesis se enfoca en el diseño y la construcción de un sistema informático que permita un mejor control de usuarios al manejo de la información relacionada con el análisis de leche, a través del uso de claves de acceso como el nombre de usuario y contraseña. Además se controlará el ingreso de usuarios a los distintos módulos u opciones, que brinde el sistema, a través del manejo de niveles de acceso aplicable a cada usuario en función del cargo y funciones que desempeñe el mismo. Esto evitará que personal no autorizado o con otras funciones ingresen a información importante y confidencial y se corra el riesgo de que la misma pueda ser modificada o eliminada.

Para llevar un registro permanente y actualizado sobre los datos personales de cada cliente, que solicita el análisis de leche, el respectivo usuario (autorizado) podrá ingresar o actualizar los datos de los clientes, indicando el tipo de cliente al que corresponda. A fin de poder procesar información y generar los resultados esperados de acuerdo a las necesidades particulares de cada cliente.

En el control automatizado de los materiales entregados, devueltos o pendientes en su entrega, para cada uno de los clientes que soliciten el servicio, se llevará un registro en el que conste la cantidad y tipo de material solicitado. Para el caso de los frascos, para la toma de muestras, se generarán el código de barras a través de una etiqueta impresa que será adherida a dichos frasco, éste código estará relacionado con la información del cliente y número de lote referente al tipo de análisis solicitado.

El proceso de recibimiento de las muestras de leche, previo a su análisis, requiere que se lleve un registro actualizado y al día, para ello el usuario autorizado podrá

registrar información como la fecha, temperatura y otras observaciones. Información que es de suma importancia, debido a que afecta directamente a los resultados finales del análisis bacteriológico y de la calidad de dichas muestras.

En el procesamiento de los resultados obtenidos, una vez que pasa el proceso de análisis en el Laboratorio de Calidad de Leche, el usuario autorizado será el encargado de seleccionar y exportar la información generada por los equipos analíticos correspondientes. La información es devuelta en archivos planos, por los equipos analíticos y los mismos serán subidos y procesados por el nuevo sistema informático.

Una vez procesada la información, referente al análisis, se procederá a la generación de reportes y/o consultas automatizadas. Esta información estará en función del tipo de cliente, es decir, que cada cliente recibirá una explicación diferente, esto se debe a que no todos los clientes alcanzan el mismo nivel de comprensión y requieren de información más explícita que técnica. Los reportes en mención, a parte, de ser generados en pantalla o en forma impresa, se tendrá la opción de poder migrarlo a archivos en otros formatos como excel, pdf, txt; con fin de poder procesarlos más fácilmente y entregarlos a los clientes de acuerdo a su requerimiento, en cuanto al formato del mismo.

La realización de gráficos estadísticos, en forma automatizada, sobre los resultados que se obtienen y de acuerdo al tipo de cliente, es otra de las opciones que proporcionará el sistema informático propuesto, a fin de poder adjuntar a los reportes de resultados para lograr una mayor comprensión de los mismos.

Para el diseño de las interfaces del sistema, se dará a conocer periódicamente a los usuarios que vayan a interactuar con el sistema mediante el diseño de prototipos, a fin de que se ajunte a los requerimientos de dichos usuarios. Adicionalmente cabe mencionar que durante todo el proceso de construcción e implementación del sistema informático se manejará altos niveles de seguridad y confidencialidad con la información que maneja cada uno de los usuarios.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Métodos instrumentales para el análisis de leche cruda

2.1.1 Bactoscan

BactoScan™ mide la calidad higiénica de la leche, contando todas y cada una de las células bacterianas presentes en la leche cruda. Al obtenerse resultados fiables en muy poco tiempo, se puede actuar rápido para preservar y mejorar la calidad de la leche (Foss-bactoscan, 1956)

2.1.2 Fossomatic

Fossomatic™ FC realiza un recuento de las células somáticas. Es un analizador de alta capacidad que le ayuda a satisfacer las demandas de los granjeros que necesitan resultados de control lechero rápidos y fiables. También se ofrece como parte de CombiFoss™ FT+ - una integración perfecta de MilkoScan™ FT+ y Fossomatic (Foss; Foss-fossomatic).

2.1.3 Milkoscan

El MilkoScan FT+ es un analizador de leche completamente automático de alta capacidad (cumple con lo establecido por la IDF y la AOAC) para el pago por calidad y control lechero. Realiza análisis de la leche cruda rápidos, fiables y precisos, con distintas opciones disponibles como el perfil de leche sana, o el análisis de ácidos grasos libres, urea o caseína (foss-milkscan)

2.1.4 Datos obtenidos de los equipos analizadores

Según la página oficial FOSS, la salida de datos de los equipos analizadores de leche requiere previamente que las muestras de leche sean registradas en el sistema informático que posee cada equipo. Seguidamente se procede a realizar el análisis de calidad de cada muestra de leche y sus resultados son manejados de la siguiente manera:

Visualización en tiempo real / impresión y el almacenamiento en el disco duro o disquete.

Transmisión del archivo, con los datos resultantes, desde el Host del equipo analizador hacia la PC que tiene el software informático de análisis de leche, por el puerto (RS232).

Exportación de datos, referente a los resultados obtenidos en el análisis, a archivos CSV, XML o protocolo CS83. (Foss)

2.2 Metodología

2.2.1 OOHDH

OOHDH (Object Oriented Hypermedia Design Method), en español se define como: Método de Diseño Hypermedia Orientado a Objetos, es una metodología utilizada para el desarrollo de aplicaciones informáticas para la Web.

Es un modelo basado en enfoque para la construcción de grandes aplicaciones hipermedia.

Se ha utilizado para diseñar diferentes tipos de aplicaciones, tales como: sitios web y sistemas de información, blogs interactivos, presentaciones multimedia, etc.

OOHDH se ha utilizado para diseñar diferentes tipos de aplicaciones, tales como sitios web y sistemas de información, blogs interactivos, presentaciones multimedia, etc.

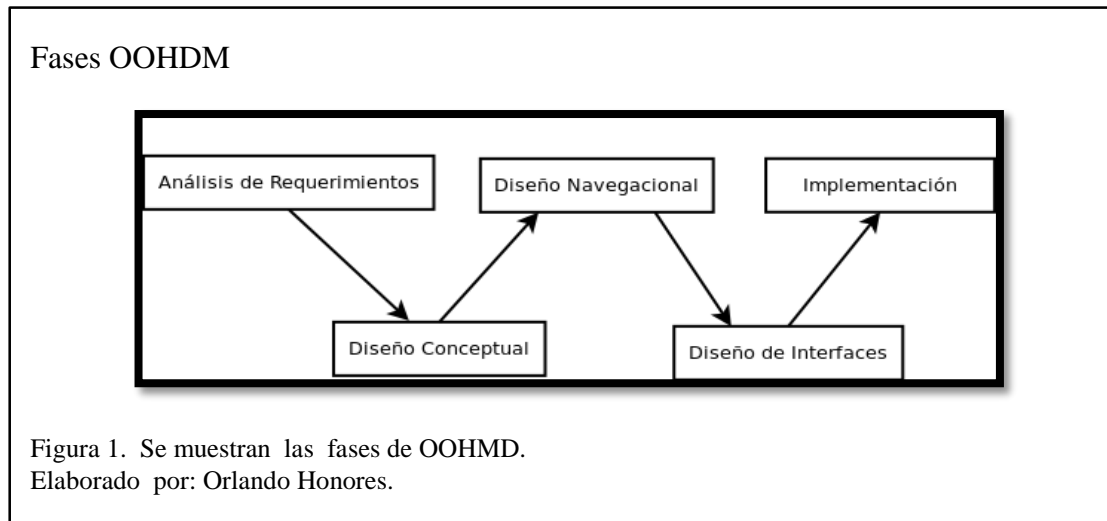
OOHDH es una combinación de estilos de desarrollo basado en prototipos, en desarrollo interactivo y de desarrollo incremental. En cada fase se elabora un modelo orientado a objetos conceptual que recoge las características a resaltar en la misma, incrementando los resaltados de la fase o fases anteriores. (Lapuente, 2013)

2.2.1 1 Fases de la metodología OOHDH

OOHDH consta de cinco actividades diferentes, a saber:

- Análisis de Requerimientos.
- Diseño conceptual.
- Diseño de navegación.
- Diseño de interfaz abstracta.

- Implementación (Scwabe, 2008)



Análisis de requerimientos

En esta fase es donde se fundamenta los diagramas de casos de uso, los cuales son diseñados por escenarios con un objetivo principal el de obtener en forma clara los requerimientos y acciones del sistema.

Es la fase principal, porque es donde se realiza la recolección de datos y para ellos se deben proporcionar algunas preguntas como:

- ¿Cuáles son los tópicos principales?
- ¿Cómo los tópicos están relacionados entre sí?
- ¿Qué categoría de usuarios serán atendidos?
- ¿Cuáles son las tareas principales que serán abordadas?
- ¿Qué tareas corresponden a los usuarios atendidos o usuarios categorizados?

Para esto, es necesario primero identificar los actores y las tareas que ellos deben realizar en casos de uso.

La siguiente figura (Figura 5.) muestra las cinco fases del análisis de requisitos de acuerdo con “Modeling Interactions and Navigation in Web Applications ”

Análisis de requerimiento

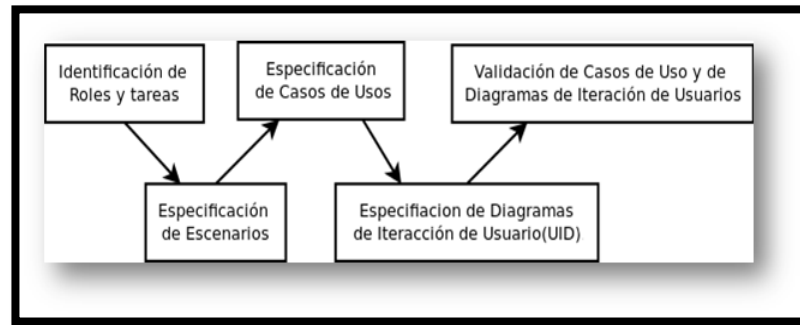


Figura 2. Las cinco fases de análisis de requerimientos.
Elaborado por: Orlando Honores.

Identificación de roles y tareas, el objetivo de esta primera fase es para diferenciar roles y actores del dominio y sus tareas relacionadas.

A continuación se detallan los cinco pasos para obtener el análisis de requerimientos de esta primera fase:

- Especificación de Escenarios, luego de haber identificado a los roles y tareas se describen los posibles escenarios.
- Especificación de Casos de Usos, los escenarios deben ser agrupados en unidades funcionales, para esto hacemos uso de UML (Casos de Uso.).
- Especificación de Diagramas de Iteración de Usuario, utiliza una herramienta llamada UID para representar la interacción entre el usuario y el sistema. UID es una herramienta que se utiliza principalmente para apoyar la comunicación entre el diseñador y los usuarios en la obtención de requisitos. Por cada caso de Uso se define un UID.
- Validación de Casos de Uso y de Diagramas de Iteración de Usuarios, el diseñador debe interactuar con cada usuario para validar los casos de uso y los UIDs generados, mientras que cada usuario valida únicamente los casos de uso y diagramas relacionados con los papeles que interpreta.

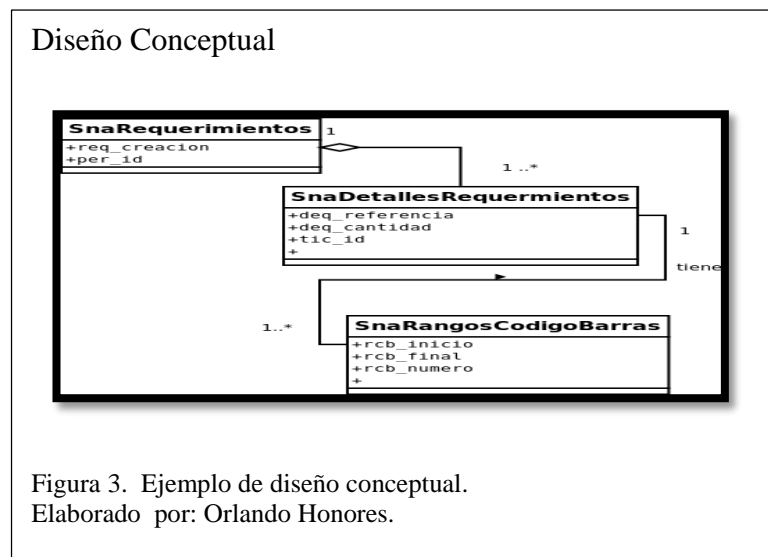
Diseño conceptual

La finalidad principal de esta fase es capturar el dominio semántico de la aplicación lo más detallado posible, teniendo en cuenta el papel de los usuarios y las tareas que desarrollan.

El resultado de esta fase es un modelo de clases relacionales, es decir que está conformado por clases, relaciones y subsistemas.

En esta fase se utiliza un lenguaje de modelado de sistemas de software llamado UML (Unified Modelling Language), que permite en forma gráfica visualizar, especificar, construir y documentar un sistema.

En la figura 3, se expone un ejemplo del Diseño Conceptual.



Diseño navegacional

En esta fase de la metodología OOHDM, se desarrolla una topología navegacional que permita a la aplicación ejecutar todas las acciones requeridas por el usuario.

La estructura de la navegación hipertexto está definida por un esquema de clases de navegación específica, que reflejan una posible vista elegida.

En OOHDM hay una serie de clases predefinidas las cuales describen la estructura de la página web, tales como clases navegaciones, nodos, enlaces, índices, menús y vistas guiadas.

A continuación se detalla algunas de estas clases predefinidas que permiten la interacción con el usuario:

Los Nodos son contenedores básicos de información de las aplicaciones hipermedia.

Enlaces, reflejan la relación de navegación que puede explorar el usuario

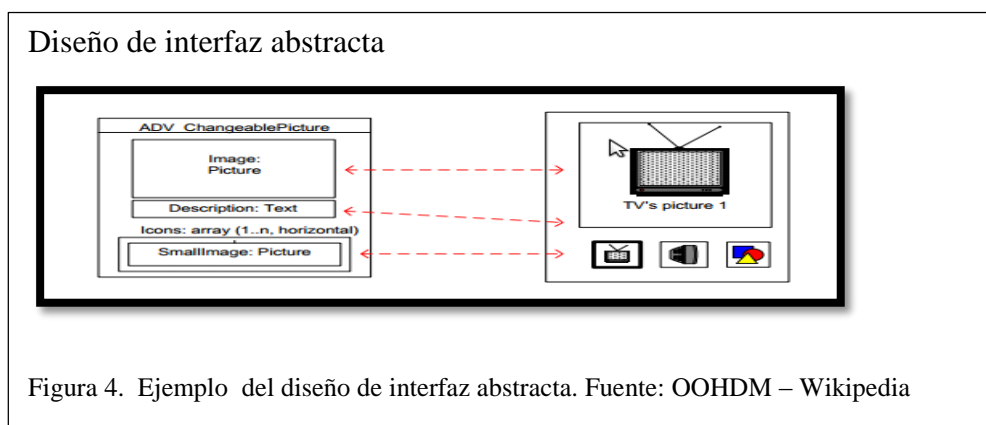
EL modelo de navegación puede evolucionar de forma independiente a partir del modelo conceptual, lo que simplifica el mantenimiento.

Estructuras de acceso, actúan como índices o diccionarios que permiten al usuario encontrar de forma rápida y eficiente la información deseada

Diseño de la interfaz abstracta

La interfaz abstracta se describe el contexto navegacional, es decir la información que deberá presentarse según la vista elegida. Esta fase permite dividir las tareas de los usuarios, así como tener diferentes interfaces para un mismo modelo navegacional.

En OOHDM se utilizan vistas abstractas de datos denominadas ADV (Abstract data views), mediante la cual se representa la estructura estática de la interfaz, la composición de objetos y los eventos a los que corresponden. En la figura 7, se puede observar el diseño estructural como su comportamiento.



Implementación

La implementación consiste en mapear los objetos del Modelo de Navegación e Interfaz a una aplicación final y puede requerir elaborar arquitecturas (por ejemplo, Cliente-Servidor) en las cuales las aplicaciones son clientes (páginas web) para

compartir el servidor de base de datos que contiene los objetos del modelo conceptual.

2.2.2 UML

El lenguaje de Modelamiento Unificado (Unified Modeling Language), es un lenguaje gráfico para visualizar, especificar, y documentar cada una de las partes de que comprende el desarrollo de software. Está conformado por un conjunto de herramientas que permite modelar (analizar y diseñar) sistemas orientados a objetos, además es independiente del ciclo de desarrollo, es decir se lo puede usar en cualquier fase del método elegido.

La versión 2 de UML posee trece tipos de diagramas divididos en tres categorías: seis tipos de diagramas representan la estructura de una aplicación; tres tipos de diagrama representan tipos generales de comportamiento; y cuatro representan diferentes aspectos de las interacciones, en la tabla 1 se muestran los diagramas establecidos en la versión 2 de UML. (Ávila Cesar, 2001) (Object Management Group, Inc, 2015)

Tabla 1.
Tipos de diagramas en UML.

Diagrama de modelo estructurado	Diagrama de clase. Diagrama de objeto. Diagrama de componentes. Diagrama de estructuras compuestas. Diagrama de paquetes. Diagrama de despliegue.
Diagramas de comportamiento	Diagrama de casos de uso, utilizado por algunas metodologías durante la recopilación de requisitos. Diagrama de actividad. Diagrama de máquina de estado, esencial para el entendimiento de un proceso en ejecución.
Diagramas de interacción	Diagrama de secuencia. Diagrama de comunicación. Diagrama de tiempos. Diagrama de descripción de la interacción.

Nota: Según la página oficial de la UML, la versión 2 cuenta con 13 diagramas
Elaborado por: Orlando Honores.

Diagramas usados en el desarrollo del software

Diagrama de casos de uso

Es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso posee uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Los diagramas de casos de uso son a menudo confundidos con los casos de uso. Mientras los dos conceptos están relacionados, los casos de uso son mucho más detallados que los diagramas de casos de uso.

En la figura 5, se describe un ejemplo de un Diagrama de Caso de Uso, mientras que en la figura 6, se muestra algunos componentes que forman un diagrama de caso de uso.

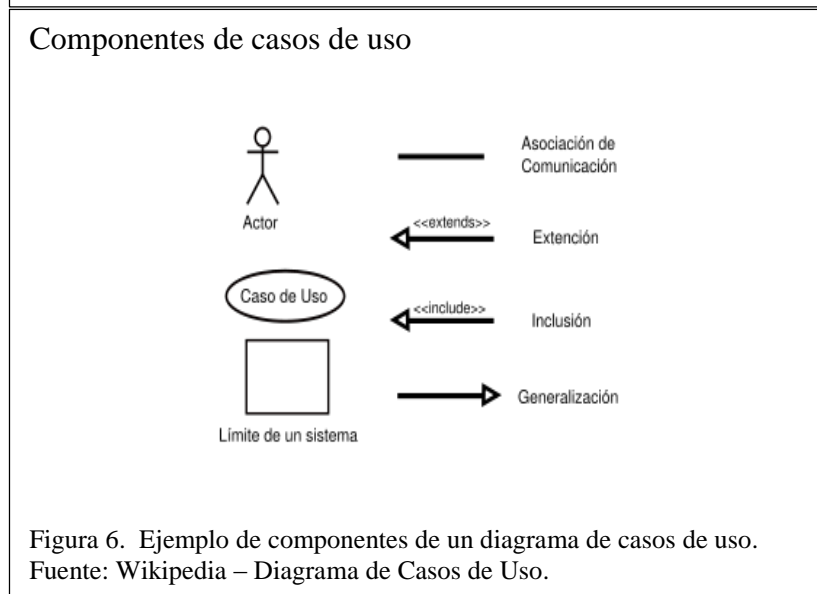
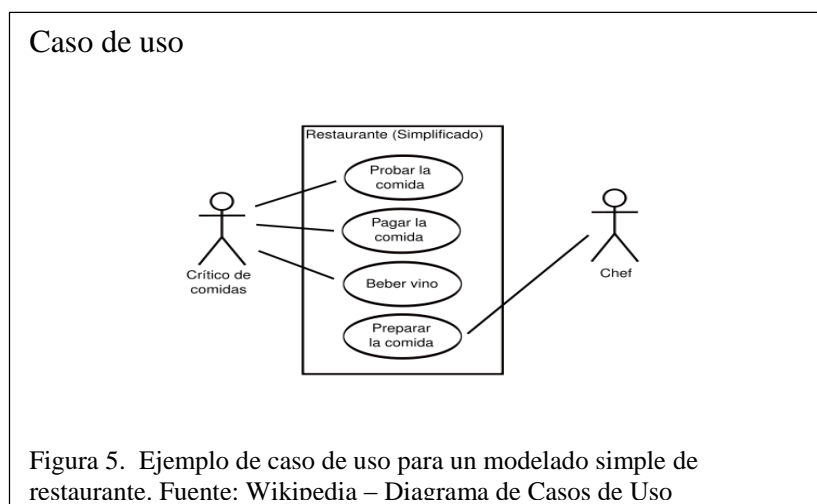


Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

En la figura 7, se detalla un diagrama de clase con sus respectivas relaciones

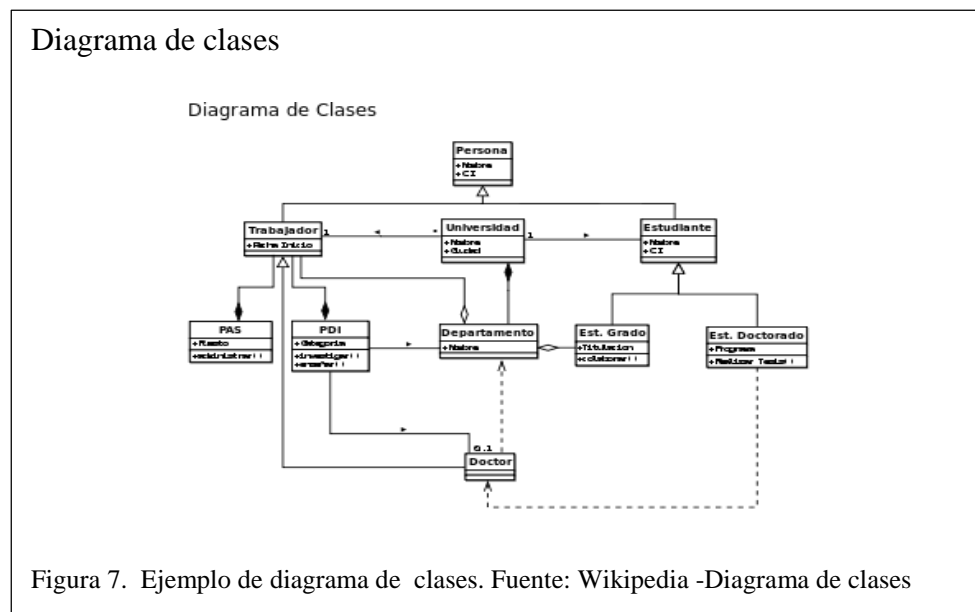


Diagrama de secuencia

El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema.

Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos. (Diagrama de secuencia, 2015)

Diagrama de Secuencia

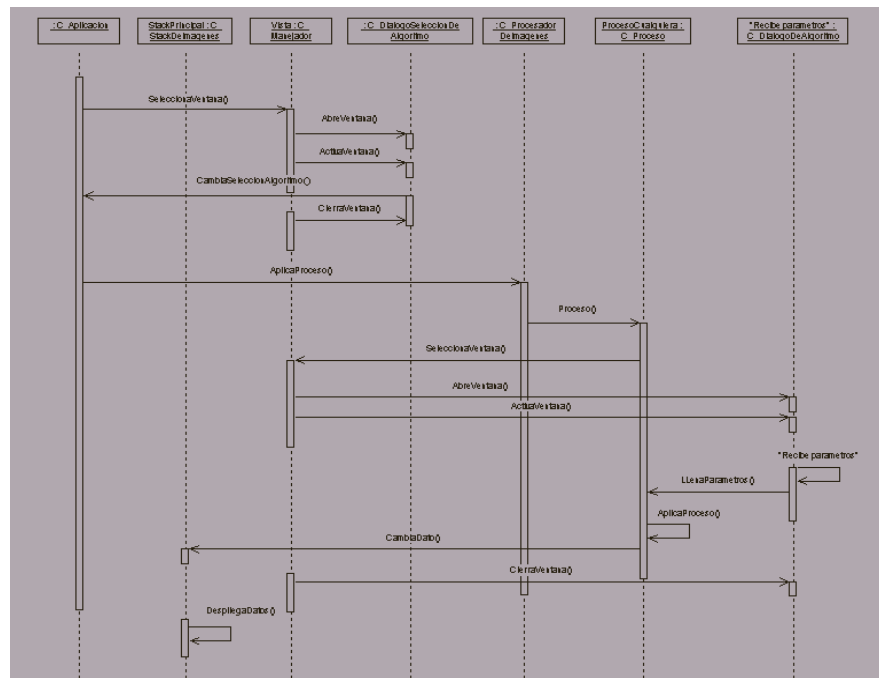


Figura 8. Ejemplo de Diagrama de secuencia. Fuente: Wikipedia – Diagrama de secuencia

Diagrama Entidad – Relación

Es una herramienta que permite representar en forma abstracta las entidades relevantes de un sistema de información así como sus relaciones y propiedades.

Es conocido en el idioma inglés como E-R "Entity relationship", mientras que en el español DER "Diagrama de Entidad Relación". (Modelo entidad-relación, 2015)

En la figura 9, se puede observar un ejemplo de modelo entidad-relación en notación UML con sus respectivas relaciones y propiedades embebidas.

Entidad –Relación notación UML

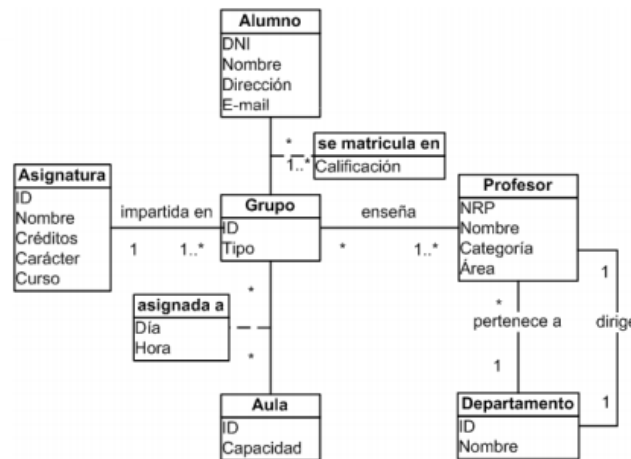


Figura 9. Ejemplo de Diagrama E-R. Fuente: Modelado de datos, página web Fernando Berzal, base de datos 2.

2.3 Herramientas de desarrollo

Todas las herramientas utilizadas para el análisis, diseño y construcción del sistema informático propuesto, en el presente proyecto, son de libre distribución por formar parte del grupo de software libre que se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software, es decir la libertad de usar el programa, con cualquier propósito; adaptarlo a las necesidades; de estudiar el funcionamiento del programa y de distribuir copias, de modo que toda la comunidad se beneficie. (hispalinux.e, 2015)

Leguaje de programación Java

Java es un lenguaje de programación orientado a objetos, que fue diseñado específicamente para trabajar en cualquier sistema operativo. Una de sus características principales es que los desarrolladores de aplicaciones creen un programa una vez y lo ejecuten en cualquier dispositivo sin necesidad de ser recompilado para correr en otra en otra máquina, esto se debe a que las aplicaciones de java se encuentran compiladas en bytecode (archivo binario que tiene un programa ejecutable), otra característica de java es la reutilización del código lo que permite a los desarrolladores ahorrar tiempo en el desarrollo de una aplicación.

A partir del 2012, es uno de los lenguajes de programación más populares. (Java (lenguaje de programación), 2015)

Java EE

Es una plataforma de programación para desarrollar y ejecutar aplicaciones en java, se ejecuta sobre un servidor de aplicaciones o de escritorio. Las varias especificaciones API que comprenden esta plataforma como JDBC, RMI, e-mail, JMS, Servicios Web , Xml, etc; permite desarrollar una aplicación de empresa portable y escalable, garantizando la seguridad, escalabilidad, concurrencia y gestión de componentes de varias tecnologías de servicio web.

La versión actual conocida es Java EE 6 que fue lanzada en el 2009, pero existe una nueva versión no oficial llamada Java EE 8, versión 2014.

Uno de los objetivos de la plataforma Java EE es acortar el tiempo a los desarrolladores mediante el conjunto de APIs, descritos anteriormente, logrando mejorar el rendimiento y reducir la de complejidad de las aplicaciones. (Java EE, 2015)

JPA

Para entender el JPA debemos tener en claro que el proceso de mapear objetos java a tablas de bases de datos y viceversa se llama ORM, Object Relational Mapping (Mapeo de Objeto Relacional).

JPA, Java Persistence API, es un enfoque posible para ORM, entonces JPA es una especificación y varias implementaciones disponibles para el mapeo de datos. Las implementaciones más populares son Hibernate, EclipseLink (referencia del JPA) y OpenJPA Apache.

A través del JPA el desarrollador puede asignar, almacenar, actualizar y recuperar datos de bases de datos relacionales a objetos java y viceversa.

La implementación JPA se suele llamar proveedor de persistencia. JPA puede ser utilizado en Java EE y aplicaciones Java SE. (GmbH, 2015)

Entity

Se denomina Entidad a una clase que representa una tabla en un modelo de datos relacional y las instancias de la clase será una fila en la tabla

En java para representar una entidad debe contener la siguiente anotación “`javax.persistence.Entity`”.

JPA creará una tabla para la entidad en su base de datos. Todas las clases de entidad deben definir una clave principal y un constructor. Las claves pueden ser un solo campo o una combinación de campos.

Ejemplo:

JPA permite generar automáticamente la clave primaria en la base de datos vía `@GeneratedValue` annotation.

Por defecto, el nombre de la tabla corresponde al nombre de la clase, la cual puede ser cambiada con la siguiente anotación: `@Tabla(name="nuevo_nombre_de_la_tabla")`. (GmbH, 2015)

Persistencia de campos.

Los campos de la Entidad se guardarán en la base de datos. JPA puede utilizar cualquiera de sus variables de instancia (campos) o los correspondientes “getters” y “setters” para acceder a los campos. JPA persiste por defecto, todos los campos de una entidad mediante la anotación `@Column`, si los campos no deben ser guardados en la base de datos deben estar marcados con `@Transient`. El nombre de la columna que viene por defecto se lo puede cambiar a través de `@Column (name = "nuevo_nombre_de_la_columna")`. (GmbH, 2015)

En la tabla 2, se describe algunas de las anotaciones más usadas:

Tabla 2.
Anotaciones en la persistencia de campos

@Id	Identifica a la entidad
-----	-------------------------

@GeneratedValue	Permite generar las clave primarias
@Transient	Campos que no pertenecen a la base de datos

Nota: Ejemplos de anotaciones. Fuente: JPA 2.0 con EclipseLink.
Elaborado por: Orlando Honores.

Relaciones entre entidades

JPA permite definir relaciones entre las clases, por ejemplo: una clase es parte de otra clase (contención). Las clases pueden tener de uno a uno, uno a muchos, muchos a uno y muchos a muchas relaciones con otras clases.

Una relación puede ser bidireccional o unidireccional, por ejemplo, en una relación bidireccional ambas clases almacenan una referencia a la otra, mientras que en un caso unidireccional sólo una clase tiene una referencia a la otra clase. (GmbH, 2015)

Entity Manager

La clase `javax.persistence.EntityManager` es la interfaz principal de JPA utilizada para la persistencia de las aplicaciones. Cada `EntityManager` puede realizar operaciones CRUD (Create, Read, Update, Delete) sobre un conjunto de objetos persistentes. (GmbH, 2015)

FrameWork JSF

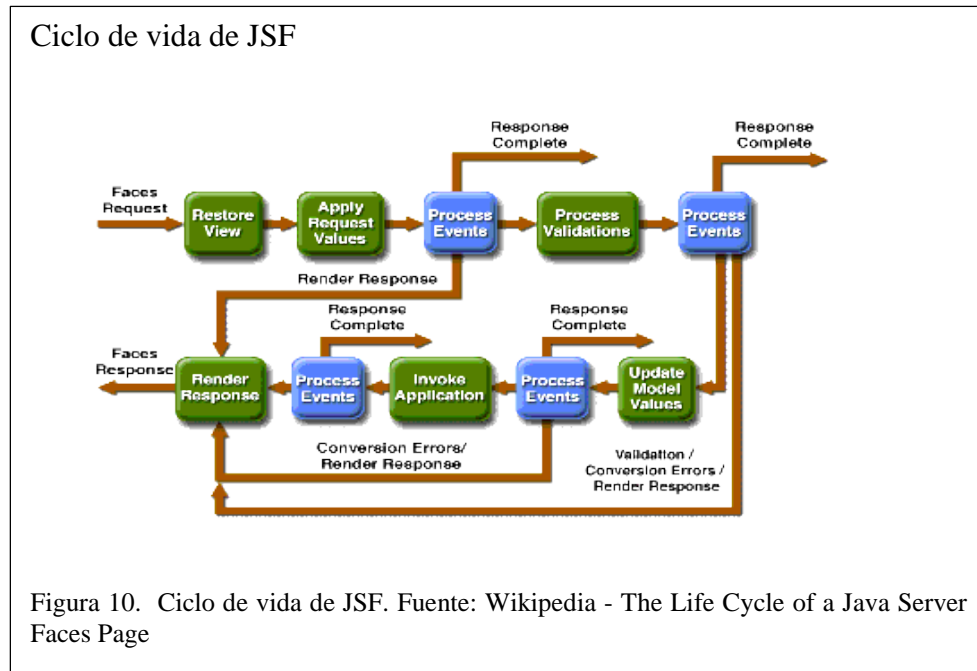
Java Server Faces, tecnología basada en java, en aplicaciones web, enfocada al MVC (Modelo Vista Controlador).

Es una especificación estándar para la construcción de interfaces de usuario (User Interface UI) para aplicaciones server-side. Antes `JavaServer Faces`, los desarrolladores quienes construyeron aplicaciones web developers a menudo se basaban en la construcción de interfaz de usuario. (GmbH, JSF (JavaServer Faces) - Tutorial, 2011)

Aplicación JSF

Una aplicación JSF se compone de páginas web con JSF componentes de interfaz de usuario. Una aplicación JSF requiere también algunos archivos de configuración ("faces-config.xml" y web.xml).

JSF utiliza el lenguaje de expresión unificada (EL) para unir los componentes de interfaz de usuario a los atributos de los objetos o métodos, su ciclo de vida está compuesto de 6 fases. El ciclo vida del JSF es una secuencia de fases por las que pasa una petición JSF desde que se recibe en el servidor hasta que se genera la página HTML resultante. El servlet que implementa el framework (javax.faces.webapp.FacesServlet) recibe la petición y realiza todo el ciclo, creando y utilizando los objetos Java que representan los componentes JSF y los beans gestionados. La relación entre estos objetos y la generación de código HTML a partir del árbol de componentes constituyen la base del funcionamiento del framework, en la figura 10, se encuentran representadas las fases. (GmbH, JSF (JavaServer Faces) - Tutorial, 2011)



Las fases del ciclo de vida son las siguientes:

- Restaurar la vista (restore view). Se obtiene el árbol de componentes correspondiente a la vista JSF de la petición y hace una actualización (render response).
- Aplicar los valores de la petición (apply request values). Una vez obtenido el árbol de componentes, se añaden todos los valores asociados a cada componente. Se convierten todos los datos de la petición a tipos de datos Java y se validan.
- Procesar las validaciones (process validations). Se validan todos los datos ingresados. Si existe algún error, se encola un mensaje de error y se termina el ciclo de vida, saltando al último paso (actualizar componentes y mostrar mensajes de errores).
- Actualizar los valores del modelo (update model values). En esta fase, todos los valores se han procesado y se han validado. Se actualizan las propiedades de los beans que tienen relación con los componentes.
- Invocar a la aplicación (invoke application). En esta fase, todas las propiedades de los beans asociados a componentes del árbol se han actualizado.
- Renderizar la respuesta (render response). Actualiza la variable en el lado del cliente (html). (Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, 2014)

AngularJs

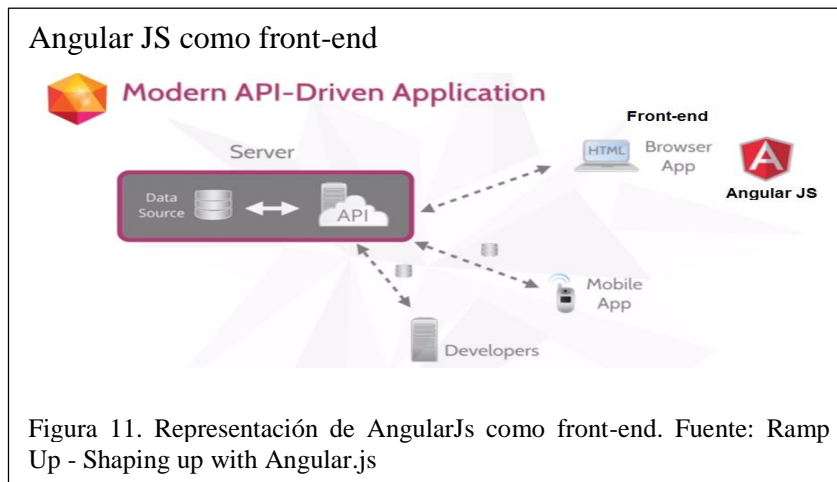
Es un framework basado en su totalidad en JavaScript de código abierto. Aplica el modelo MVC en una página web, lo cual permite el control total de la misma.

La programación se encuentra en el lado del cliente (controlador), que mediante la implementación de Ajax permite la comunicación con el servidor mediante servicios.

Permite la creación de HTML enriquecido, insertar en forma dinámica código JavaScript y html, lo que hace que la programación sea reusable.

Angular permite repartir la lógica de la aplicación por capas, lo que resulta muy adecuado para aplicaciones de negocio y para las aplicaciones SPA (Single Page

Application). La figura 11., representa el uso de AngularJs como front-end (parte del software que interactúa con los usuarios)



Una aplicación AngularJS está compuesta por módulos, dependencias, modelos, controladores y servicios; esto permite que el código sea mantenible, testeable y legible. (Code School, 2015)

La siguiente figura 12., representa la creación de un módulo en AngularJS



Socket IO

Es una librería creada en JavaScript para Node.js que permite una comunicación bidireccional en tiempo real entre el cliente y el servidor. Se basa principalmente en

WebSocket, pero también se base en alternativas como json polling o long polling en Ajax.

Trabaja como un socket en java, pero dedicado a la web en donde es posible enviar datos binarios. La implementación es muy sencilla en ambos lados (servidor-cliente), la cual ha sido muy usada en la actualidad en proyectos que forman parte de:

- Análisis en tiempo real.
- Mensajería instantánea y chat.
- Transmisión binaria.
- La colaboración de documentos.

En marzo del 2015 se lanzó la versión 1.3.5, esta versión cuenta con un (socket.io, 2015)

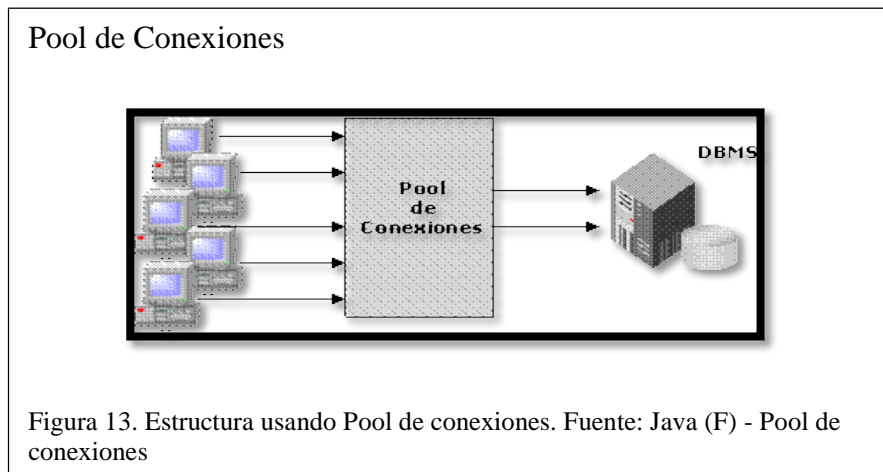
2.4 Pool de conexiones con la base de datos

Es un conjunto limitado de conexiones a una base de datos, administrado por un servidor de aplicaciones, que permite la reutilización de las mismas, es decir que va asignando las conexiones a medida que los clientes realicen consultas o actualicen información de la base de datos.

La ventaja principal de las aplicaciones que utilicen una pool de conexiones frente a una arquitectura tradicional de cliente/servidor (2 capas) es que el acceso a la conexión a la base de datos no se hace desde el cliente, como en una aplicación de 2 capas, sino que es centralizado y controlado mediante el servidor de aplicaciones.

El manejo de una pool favorece la escalabilidad y performance de una aplicación porque evita que se consuman recursos, memoria y tiempo del servidor de la base de datos por cada nueva conexión. (Artech Consultores S.R.L., 2013)

En la figura 13, se muestra una pool de conexiones en forma abstracta



2.5 Eclipse como Ide, Entorno de desarrollo integrado

Eclipse es un programa multiplataforma compuesta por uno o más extensiones (plugins) de código abierto, usada para desarrollar entornos de desarrollo integrados (IDE), compuesta principalmente de JDT(Java Development Toolkit) y el compilador ECJ (Eclipse Compiler for Java).

Algunas de sus características principales es su editor de texto con un analizador sintáctico enriquecido, compilación en tiempo real, pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes para creación de proyectos, clases, etc., y refactorización. (Eclipse (software), 2015)

2.6 Servidores

2.6.1 Node.js

Node.js® es una plataforma construida en tiempo de ejecución de JavaScript de Chrome para construir fácilmente aplicaciones rápidas de red escalables. Node.js es basado en eventos asíncronos que lo hace ligero y eficiente, ideal para aplicaciones en tiempo real de datos intensivos que se ejecutan a través de dispositivos distribuidos. (Node.js Foundation, 2015)

2.6.2 Apache Tomcat

Es un contenedor web con soporte de servlets y JSPs, escrito totalmente en java, lo que permite que funcione en cualquier sistema operativo que disponga de la máquina virtual de java.

Se lo conoce como simplemente Tomcat, en sus inicios se pensaba que dicho servidor era recomendable usarse en entornos de desarrollo con requisitos mínimos de velocidad. En la actualidad no existe esta percepción y por esto, es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Las principales características que soporta la última versión 7.x estable son:

- Autenticación de acceso básico.
- Negociación de credenciales.
- HTTPS
- Alojamiento compartido.
- CGI o interfaz de entrada común.
- Servlets de Java.
- SSI
- Consola de administrador.

Con estas características se convierte en uno de los servidores más usados. (Tomcat, 2015)

2.6.3 Postgresql

Es un SGBD relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan en libre por organizaciones comerciales.

Algunas de sus principales características son, entre otras:

Alta concurrencia

Mediante un sistema denominado MVCC (Acceso concurrente multiversion, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Amplia variedad de tipos nativos

PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas).
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Otras características:

Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (foreign keys).

Disparadores (triggers): Un disparador o trigger se define como una acción específica que se realiza de acuerdo a un evento, cuando éste ocurra dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Todos los disparadores se definen por seis características:

- El nombre del disparador o trigger
- El momento en que el disparador debe arrancar
- El evento del disparador deberá activarse sobre...
- La tabla donde el disparador se activará
- La frecuencia de la ejecución
- La función que podría ser llamada

Entonces combinando estas seis características, PostgreSQL permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (triggers).

Tipos de datos y operaciones geométricas.

El soporte para transacciones distribuidas permite a PostgreSQL integrarse en un sistema distribuido formado por varios recursos (p.ej, una base de datos PostgreSQL,

otra Oracle, una cola de mensajes IBM MQ JMS y un ERP SAP) gestionado por un servidor de aplicaciones donde el éxito ("commit") de la transacción global es el resultado del éxito de las transacciones locales.

Funciones

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional. (PostgreSQL, 2015)

2.7 Reportes

Un reporte es un documento que presenta de manera estructurada y/o resumida, datos relevantes guardados o generados una aplicación, generalmente agrupan los datos de acuerdo a un interés específico.

2.7.1 JasperReports

JasperReports es una herramienta de creación de informes que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML.

Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico.

Su propósito principal es ayudar a crear documentos de tipo páginas, preparados para imprimir en una forma simple y flexible.

JasperReports se usa comúnmente con iReport, un front-end gráfico de código abierto para la edición de informes. Se encuentra bajo licencia libre GNU, por lo que es Software libre. (JasperReports, 2015)

2.7.2 IReports

IReport es un diseñador gratuito y de código abierto para JasperReports. Crea diseños muy sofisticados que contienen gráficos, imágenes, sub-informes, tablas de referencias cruzadas y mucho más. Puede acceder a datos a través de JDBC, TableModels, JavaBeans, XML, Hibernate, CSV, y fuentes personalizadas y luego

publicar estos informes en formato PDF, RTF, XML, XLS, CSV, HTML, XHTML, texto, DOCX, u OpenOffice. (TIBCO Software, Inc, 2015)

2.8 Sistema operativo

2.8.1 Ubuntu

Es un sistema operativo basado en GNU/Linux y que se distribuye como software libre. Es conocido por su facilidad de uso en sus las aplicaciones, entre sus características principales en la última versión es el soporte oficialmente a dos arquitecturas de hardware en computadoras personales y servidores 32-bit y 64-bit, es capaz de actualizar a la vez todas las aplicaciones instaladas en la máquina a través de repositorios.

Su última versión con soporte es la 15.04 con el nombre clave Vivid Vervet lanzada en abril del 2015, para el 2016 se viene la versión 15.10. (Ubuntu, 2015)

2.8.2 Microsoft Windows

Microsoft Windows es el nombre de una familia de sistemas operativos desarrollados y vendidos por Microsoft.

Microsoft introdujo un entorno operativo denominado Windows el 20 de noviembre de 1985 como un complemento para MS-DOS en respuesta al creciente interés en las interfaces gráficas de usuario (GUI). Microsoft Windows llegó a dominar el mercado mundial de computadoras personales, con más del 90% de la cuota de mercado, superando a Mac OS, que había sido introducido en 1984.

Las versiones más recientes de Windows son Windows 8 para equipos de escritorio, Windows Server 2012 para servidores y Windows Phone 8 para dispositivos móviles. (Microsoft Windows, 2015)

CAPÍTULO 3

ANÁLISIS Y DISEÑO DE PROCESOS

3. 1 Análisis de la situación actual

Actualmente el Laboratorio de Calidad de Leche, cuenta con un software (Qualion) de procedencia brasileña y que es utilizado por los laboratorios de la Red Brasileira de Laboratorios de Controle de Qualidade do Leite – RBQL.

El proceso de análisis de calidad de la leche, llevada a cabo por equipos tecnológicos especializados cuyos resultados obtenidos son procesados por el equipo técnico que labora en el Laboratorio de Leche, se basa en el estudio de los siguientes parámetros:

- Composición: se determina grasa, proteína, lactosa, sólidos totales y sólidos no grasos con base en métodos de análisis infrarrojos (MilkoScan FT 6200-Foss).
- Contaje de células somáticas: se cuentan las células somáticas empleando un método fluoróptico (Fossomatic Minor- Foss).
- Contaje total de bacterias: se determinan el número de unidades formadoras de colonias de bacterias con métodos fluorópticos-electrónicos (BactoScan 50-H Foss).

Actualmente, todos los equipos especializados disponibles en el laboratorio de leche, emiten los resultados obtenidos del análisis de todas las muestras de leche a través de la red interna hacia al equipo principal, donde se encuentra instalado el software Qualion para el procesamiento y registro de estos resultados.

Ante las limitaciones que tiene el software Qualion en la emisión de resultados finales, debido a que solo registra los resultados más no realiza ninguna evaluación de los mimos, ha hecho que esta tarea sea ejecutada manualmente por la persona responsable, demandando tiempo y esfuerzo. Todo esto ha ocasionado que los informes finales de los resultados obtenidos sean emitidos, a los usuarios o productores de leche del Laboratorio, con retrasos.

3.1.1 Levantamiento de procesos

Para el análisis de muestras de los diferentes productores o clientes del Laboratorio de Leche de Cayambe e realiza los siguientes procesos:

Proceso 1

Registro de nuevos clientes y productores, los requerimientos de este proceso son:

- Se pide los datos personales del cliente
- Se pide los datos personales de los productores con los que cuenta el cliente.
- Se registra en el sistema primero el cliente y luego los productores

Proceso 2

Registro de pedidos de requerimientos solicitados por los clientes, los requerimientos de este proceso son:

- Se crea en forma manual el pedido de requerimientos indicando los análisis a realizar.
- Si el cliente posee una lista de las vacas a realizar los análisis con información adicional para el código de barras, se crea un archivo Excel en donde se indican que columna representan las vacas y que columna es la información que se imprimirá en el código de barras.
- Se hace un upload al sistema del archivo previamente creado. El sistema solo acepta archivos de tipo Excel con una plantilla ya definida por los creadores del sistema. Se escoge en el sistema los registros ingresados previamente por el archivo Excel, indicando los conservantes a entregar.
- Se entrega al cliente el pedido para que se acerque a cancelar en la tesorería de la UPS. El pedido lo realizan en forma manual, es decir lo que hicieron en el sistema lo vuelven hacer pero en forma manual o viceversa.

Proceso 3

Entrega de materiales, los requerimientos de este proceso son:

- Una vez cancelado el pedido de requerimientos, se entregan los materiales.

- Se imprimen los códigos de barra, el sistema imprime un código de barra por vaquita o productor, y el mismo código de barras es repetido según el número de conservantes que se haya ingresado.
- Se registra en forma manual, los frascos y conservantes entregados. El sistema no permite un control de material entregado
- Se entrega al cliente un recibo de los materiales entregados.

Proceso 4

Recepción de muestras, los requerimientos de este proceso son:

- Se reciben las muestras de los clientes.
- Se cuenta el total de muestras y confirman con el sistema.
- Se crea un nuevo lote por cliente en el sistema.
- Se lleva un registro manual de las muestras recibidas, este conjunto muestras se le denomina “Colecta”
- Se registra en forma manual y en el sistema las colectas recibidas.
- Por cada colecta se ingresa el número de lote, fecha de recepción, fecha de colecta, temperatura y observaciones.
- Se detallan las observaciones que existan por colectas.
- Si una muestra no tiene código de barras, el sistema permite reimpresiones.
- Si el cliente entrega muestras que no constaban en el requerimiento, se crea un nuevo requerimiento por esas muestras.

Proceso 5

Proceso de análisis del lote, los requerimientos de este proceso son:

- Una vez que el cliente ha entregado todas las muestras o según el tiempo transcurrido (1 hora de espera) se lleva a cabo el análisis del lote.
- Se cierra el lote en forma manual e indicando al sistema que no se puede ingresar más colectas al lote de cliente x.
- Se traslada el lote hacia los diferentes analizadores (Cfr. Supra). Los analizadores aceptan un nombre único por cada lote.

- Se ingresa como nombre del lote el código del lote.
- Se procesa el lote.
- Una vez que finalice el proceso de análisis, los analizadores arrojan un documento en formato csv.
- El resultado en formato csv es grabado en un servidor.
- El administrador envía el archivo csv al sistema.
- Si el analizador no entrega un archivo csv, se procede a crear un archivo en Excel con los resultados obtenidos, haciendo relación el resultado con el código de barras.
- El laboratorista ingresa al sistema los resultados provenientes del archivo Excel.
- Si el sistema encuentra un código de barras que no pertenece al lote, no procesa el archivo.
- Si existe un error al procesar el archivo el administrador debe solucionar el problema. Para esto debe identificar el frasco que no ha sido recibido. Lamentablemente el sistema no devuelve el número de código de barras que no pertenece al lote, solo envía un mensaje de error.
- Una vez identificado el frasco, el administrador lo elimina del archivo csv y vuelve a subirlo al sistema.
- El sistema procesa el archivo.

Proceso 6

Entrega de resultados, los requerimientos de este proceso son:

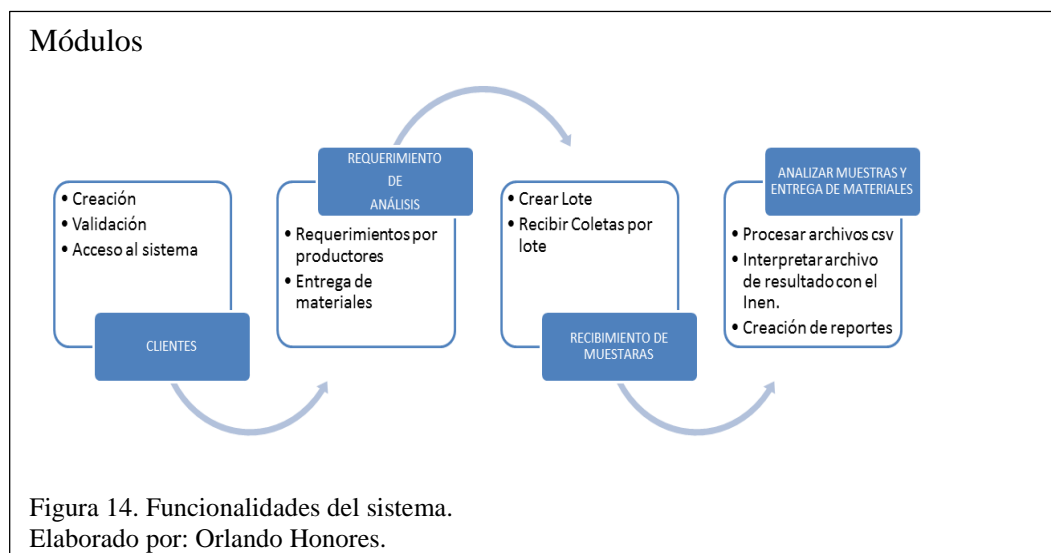
- Una vez procesado el archivo, el sistema hace una distribución de los resultados mediante el código de barras.
- Se busca el requerimiento por medio del cliente.
- El sistema muestra los resultados.
- El administrador puede exportar los resultados en un archivo Excel.
- El administrador debe aumentar nuevas variables al archivo como son los promedios, media, máximo y mínimo por análisis.
- El administrador da un formato al archivo de resultados.

- El administrador crea gráficas por medio de Excel.
- El administrador hace una comparación con las normas del INEN.
- El administrador hace una comparación con resultados anteriores, ya sea mensual o anual.
- El administrador entrega el archivo al personal calificado del QA para verificar que los resultados no se hayan alterado.
- El administrador imprime el resultado y se lo entrega al recepcionista.
- El recepcionista entrega el resultado al cliente.
- El sistema no realiza estadísticas.
- El sistema solo se encarga de guardar los resultados pero obtiene promedios por análisis y no crea gráficas.
- Cliente recibe resultados.

3.1.2 Análisis de requerimientos del sistema

El objetivo de este punto es dar a conocer a los usuarios del sistema del Laboratorio del Leche de Cayambe de la Universidad Politécnica Salesiana y al developer del proyecto tener una amplia perspectiva de las necesidades y requerimientos que posee el Laboratorio referente al análisis de muestras de leche.

Módulos



Para una mejor comprensión de los requerimientos funcionales del software propuesto se ha organizado, el mismo, en módulos (Ver figura 14)

A continuación se describe los módulos descritos anteriormente.

Módulo cliente

En este módulo se podrá crear, visualizar, editar y dar de baja los clientes.

El Laboratorio de Calidad de Leche de Cayambe cuenta con varios clientes:

- Empresas grandes que trabajan con varios productores.
- Campesinos dueños de unas pocas vaquitas.
- Productores de leche independientes que están formados por varias vacas.

Después de un análisis exhaustivo se crearon tres tipos de clientes:

- Persona
- Empresa
- Productor

Todos los tipos de clientes están relacionados por el “alias”, palabra clave para identificar al cliente, esto permitirá conexiones entre clientes, es decir una empresa puede contar con un registro permanente y actualizado sobre los datos personales de cada productor y viceversa un productor puede pertenecer a varias empresas.

La validación del cliente se la hace mediante servicios creados en servidor NodeJS que se conecta al SRI y CNE para verificar que la identificación exista y obtener el nombre o razón social del cliente o usuario.

Módulo requerimientos y entrega de materiales

En este módulo se podrá crear, visualizar y dar de baja los distintos requerimientos solicitados por el módulo “Clientes”.

Los Requerimientos son los “Análisis” que ofrece el Laboratorio de Calidad de Leche de Cayambe.

Este módulo se encuentra dividido en dos partes:

- Nuevo
- Entrega

Por cada nuevo requerimiento se generarán códigos de barra según el número análisis y de vacas a examinar.

Las etiquetas representan el código de barras que es único por vaquita, pero se repite por el número de frascos entregados por vaquita.

Cuando un cliente hace un pedido de análisis sigue dos pasos:

Solicita el requerimiento (Nuevo), y luego de ser procesado el cliente puede retirar los materiales (Entrega).

Existirá una sola pantalla para el ingreso de requerimiento o entrega de materiales, pero que se dividirá al momento de escoger el tipo de transacción a realizar.

Al finalizar la transacción en los dos casos, el laboratorista contará con un reporte del pedido del requerimiento o la entrega de materiales o en formato PDF con la finalidad de entregarle al cliente.

Módulo recibimiento de muestras

En este módulo se podrán realizar las siguientes tareas:

- Crear, visualizar y dar de baja los lotes.
- Ingresar los las colectas por lotes.
- Confirmar el análisis que se realizará al lote.
- Ingresar los las observaciones por colecta.

Lote, es cualquier cantidad de material de características similares, provenientes de una fuente común (Cliente).

Colecta, es el conjunto de unidades de muestras que conforman un lote. Se las recibe siempre y cuando el lote no ha sido analizado en caso contrario se creará un nuevo lote.

Una vez hecho el requerimiento y luego de haberse entregado los materiales, el cliente llevará las muestras de los diferentes productores al Laboratorio de Calidad

de Leche de Cayambe, donde un laboratorista encargado de recibir las muestras atenderá al cliente, todo esto proceso se lo conoce como “Recibimiento de muestras”.

Existirá una pantalla para la creación del lote, confirmación de análisis e ingreso de colectas y observaciones.

El laboratorista para la creación de un lote, tiene la opción de ingresar el código de barras en forma manual o usando el lector de código de barras.

Luego de haber recibido las colectas el laboratorista cerrará el lote y enviará las muestras a los analizadores.

Módulo resultado

En el módulo “Resultados” se podrá registrar y procesar, en forma automatizada, los resultados obtenidos de las muestras de leche, una vez que pasa el proceso de análisis en el Laboratorio de Calidad de Leche.

Se generarán reportes por requerimientos de clientes en formato WEB, PDF y XLS, los cuales incluirán gráficos estadísticos, en forma automatizada, sobre los resultados que se obtiene en cada muestra de leche analizada, de acuerdo al tipo de cliente y considerando los históricos por cliente.

Para este módulo se creará una aplicación jar en java independiente del sistema, que se encargará de llevar los archivos en formato csv emitidos por las diferentes máquinas con las que cuenta el Laboratorio de Calidad Leche hacia el servidor del sistema en forma automatizada. Pero a su vez el sistema contará con una opción de subir los archivos en forma manual.

Módulo mantenimiento

En este módulo se podrá crear y editar varios parámetros, los cuales son utilizados en diferentes módulos del sistema, los parámetros a ser administrados se describen a continuación.

Varios tipos:

- Configuración de pantallas
- Inen

- Menús
- Archivos
- Usuarios

3.2 Diseño

3.2.1 Requerimientos funcionales

Son aquellos que fueron tomados a partir de las necesidades en el Laboratorio de calidad de leche de Cayambe de la Universidad Politécnica Salesiana y que fueron automatizados por el sistema propuesto, se detalla a continuación los procesos:

Proceso 1

Autenticación de usuarios, el sistema debe solicitar credenciales como nombre y clave para poder iniciar sesión.

Requerimientos:

- Realizar una pantalla que solicite el nombre y clave del usuario.
- Indicar cuando si la clave o usuario fueron ingresados incorrectamente.
- Iniciar sesión, permitir ingresar al sistema.

Proceso 2

- Parametrización, permitir que en el sistema se pueda parametrizar los datos con los que se van a trabajar al momento de subir un archivo, en las validaciones de tipos de identificación, en la creación de nuevos menús, en la asignación de permisos, en la gestión de nuevos ítems

Requerimientos:

- Realizar una pantalla donde se puedan ver, ingresar, modificar o eliminar datos de constantes como tipos de identificación, tipos de clientes, tipos de perfiles, tipos de observaciones, tipos de requerimientos (si es nueva requisición o es entrega de materiales) y tipos de acciones que puede hacer el usuario. Los campos de entrada son nombre y descripción.
- Realizar una pantalla en la cual se pueda introducir los datos necesarios de la tabla del Inen. (Cfr. Supra).

- Realizar una pantalla para ingresar los diferentes tipos de análisis que ofrece el Laboratorio de Leche de Cayambe, donde se pueda ver, añadir, modificar y dar de baja.
- Realizar una pantalla que trabaje junto con los tipos de análisis ya que cada análisis están conformados por conservantes, y de ser posible hacer una relación de los componentes con los conservantes.
- Realizar una pantalla donde se pueda especificar el formato de archivos a recibir ya sea de clientes o de los analizadores (Cfr. Supra)
- Permitir la configuración de archivos de analizadores, permitiendo ingresar todas las columnas con las que está formado el archivo, las columnas de donde se extraerán solo resultados específicos, que columna hace referencia con el código de barras, el formato del nombre y un identificador ya sea un color que me permita ver si el archivo fue subido correctamente.
- Permitir la configuración de archivos de clientes, permitir ingresar las columnas de las que estará formado el archivo o archivos, la columna que se imprimirá con el código de barras, formato del archivo.

Proceso 3

Administrar clientes, permitir que en el sistema se pueda gestionar los clientes de manera que se pueda ingresar, modificar o darlo de baja

Requerimientos:

- Realizar una pantalla donde se pueda ingresar los clientes.
- Realizar una pantalla donde se pueda ingresar los productores.
- El sistema también deberá permitir el ingreso de empresas o personas.
- Los clientes deben contar de al menos con el ingreso de nombres, apellidos, identificación, alias, dirección (ciudad y calle), teléfono e email.
- Si un cliente es una empresa, la pantalla de ingreso de datos debe de contener ruc, nombre, alias, dirección (ciudad y calle), teléfono e email.
- El sistema debe tomar en cuenta que un cliente puede tener varias empresas y empresa puede tener un representante.
- El sistema debe permitir que un cliente tipo “persona” y “empresa” puede tener varios productores.

- El sistema debe permitir que un productor, empresa y persona pueda tener varias vaquitas.
- Realizar una pantalla donde se puede ver el número de productores que cuenta un cliente y viceversa.
- Realizar una pantalla donde se puede ver los requerimientos por clientes.
- Realizar una pantalla donde se puede editar los clientes.
- Realizar una pantalla donde se puede editar las empresas.
- Realizar una pantalla donde se puede editar los productores.
- Realizar una pantalla que permita dar de baja a los clientes.
- Realizar una pantalla que permita dar de baja a los productores.
- Realizar una pantalla que permita dar de baja a las empresas.
- Validar el ingreso de los clientes, hay clientes que tienen ruc y hay otros clientes que no tienen ruc.

Proceso 4

Requisición de análisis y entrega de materiales, permitir que en el sistema se pueda gestionar la recepción de pedidos de análisis y entrega de materiales

Requerimientos:

- Realizar una pantalla que permita ver los requerimientos activos, pendientes por entregar materiales o resultados de los análisis.
- Permitir hacer una búsqueda por cédula, nombre, alias o estado
- Realizar una pantalla que permita el ingreso de nuevos requerimientos. En la cual el usuario pueda ingresar el contacto, observaciones, subir el archivo de las vacas a realizar el análisis. Esta pantalla debe permitir escoger, editar o eliminar los análisis por vaca. Las entradas para esta pantalla son Análisis y vaquitas, es decir que permita que una vaca pueda escoger varios análisis y seguidamente que una vaca pueda tener un solo análisis.
- El sistema debe indicar la cantidad de materiales y conservantes a entregar con solo escogiendo el análisis.
- Permitir ver el total a cancelar.
- Permitir ver el pedido en formato pdf para impresión posterior.

- El sistema debe permitir realizar un cambio si es que no se ha confirmado el pedido.
- Realizar una pantalla donde se pueda ver el estado de un requerimiento, a que cliente pertenece.
- Realizar una pantalla donde se pueda ingresar los materiales entregados. Debe permitir escoger el cliente, el requerimiento del cual se le entregarán los materiales. En esta pantalla debe permitir escoger el tipo de análisis, los conservantes que pertenecen al análisis y permitir el ingreso de la cantidad entregada.
- Permitir ver el saldo de los materiales pendientes por entregar.
- Permitir ver la entrega en formato pdf para impresión posterior.
- Permitir ver un historial de los materiales entregados en diferentes formatos para luego ser exportados.
- Realizar una pantalla donde se pueda dar de baja a los requerimientos que solo tenga acceso el administrador.
- Permitir ver los resultados del requerimiento.
- Permitir ver los las gráficas de los resultados.
- Permitir exportar los requerimientos en diferentes formatos.
- Permitir exportar los resultados en diferentes formatos.

Proceso 5

Recepción de muestras, permitir que en el sistema se pueda gestionar los lotes recibidos, permitiendo el ingresar las colectas por lote y cliente.

Requerimientos:

- Realizar una pantalla que permita ver los lotes con su respectivo estado.
- Permitir hacer una búsqueda por lote, alias o estado.
- Realizar una pantalla que permita el ingreso de nuevos Lotes. En la cual el usuario pueda ingresar el código de barra por medio del lector de código de barras o ingreso manual, la persona que entrega la colecta, la fecha de colecta, fecha de recepción de colecta, la temperatura y la cantidad.
- Permitir ver los materiales entregados.
- Validar el número de muestras recibidas.

- Permitir el ingreso de observaciones ya sea por código de barras o grupal.
- Permitir la confirmación del análisis requerido.
- Realizar una pantalla donde se pueda ver el lote. Permitiendo ver un historial de las colectas recibidas, la confirmación de análisis y las observaciones.
- Permitir que solo el administrador elimine al lote,
- Permitir cerrar el lote. Ya sea manual o cuando el número de muestras recibidas sea igual al número de material entregado.

Proceso 6

Subir archivos de vaquitas a las cuales se les realizará los análisis, permitir que en el sistema se pueda subir archivos en forma manual.

Requerimientos:

- Realizar una pantalla que permita ver los archivos subidos al sistema.
- Permitir hacer una búsqueda por nombre del archivo.
- Realizar una pantalla que permita el subir archivos de clientes. En la cual el usuario pueda ingresar el alias del cliente la referencia del formato del archivo.
- Si el archivo no se puede subir permitir un hacer manualmente.
- Realizar una pantalla donde se puede ver el archivo subido.
- Permitir añadir, editar o dar de baja los ítems del archivo.
- Permitir que solo el administrador elimine los archivos de clientes.

Proceso 7

Subir archivos provenientes de los analizadores., permitir que en el sistema se pueda subir archivos en forma manual o automática.

Requerimientos:

- Realizar una pantalla que permita ver los archivos subidos al sistema.
- Permitir hacer una búsqueda por nombre del archivo.
- Realizar una pantalla que permita el subir archivos de resultados. En la cual el usuario pueda escoger el tipo de referencia del archivo es decir de donde proviene que puede ser de los analizadores (Cfr. Supra).

- Si el archivo no se puede subir permitir un hacer manualmente.
- Realizar una pantalla donde se puede ver el archivo subido.
- Permitir añadir, editar o dar de baja los ítems del archivo.
- Realizar un proceso donde los archivos generados por los analizadores se agreguen al sistema automáticamente. Si el archivo fue subido dar una alerta o cambiar el nombre del archivo. En caso contrario indicar por que no fue subido.
- Validar la información del archivo e indicar cuáles códigos de barra no pertenecen al lote o al sistema.

Proceso 8

Gestionar códigos de barra, permitir que en el sistema se pueda generar códigos de barra según los requerimientos del cliente.

Requerimientos:

- Realizar una pantalla que permita ver los rangos de códigos de barra.
- Permitir hacer una búsqueda por requerimiento o alias del cliente.
- Realizar una pantalla que permita ver los códigos de barra por rango. La misma que permitirá imprimir.
- Realizar una pantalla que permita generar códigos de barra para uso interno del Laboratorio y que el número del código de barra sea proporcionado por el cliente o en forma automática.
- Impedir que estos códigos de barra generados en forma manual por el cliente interfieran con los códigos de barra generados al momento de realizar un requerimiento.
- Realizar una pantalla donde se listar todos los códigos de barra sin rangos.
- Permitir hacer una búsqueda por requerimiento o alias del cliente o por el número de códigos de barra.
- Realizar una pantalla donde se pueda ver un solo código de barra y poder imprimirlo.

Proceso 9

Mantenimiento, permitir que en el sistema se pueda agregar nuevos usuarios, dar permisos y gestionar el menú del sistema.

Requerimientos:

- Realizar una pantalla que permita ver los usuarios de sistema.
- Permitir hacer una búsqueda por alias.
- Realizar una pantalla donde permita el ingreso de nuevos usuarios. Esta pantalla estará formada de los mismos campos cuando se ingresa un cliente, pero con la diferencia que permita ingresar la clave del usuario.
- Realizar una pantalla que permitir dar acceso al sistema a los usuarios previamente creados ya sea a usuarios internos o externos como clientes y asignar el perfil.
- Realizar una pantalla donde puede ver la información del usuario del sistema.
- Realizar una pantalla donde puede editar y dar de baja usuarios del sistema.
- Realizar una pantalla donde se pueda ver los menús del sistema.
- Permita hacer búsqueda por nombre del menú, detalle o url.
- Realizar una pantalla donde se pueda ingresar nuevos menús. Permitiendo el ingreso del nombre y descripción.
- Permitir que el ingreso del menú sea automatizado.
- Permitir al usuario administrador subir un archivo ya sea imagen, video o texto que desea mostrar al personal o a los clientes mediante el menú del sistema.
- Permitir que el usuario administrador cree sus propias blogs.
- Realizar una pantalla donde se pueda ver el menú previamente creado.
- Realizar una pantalla que permita editar el menú.
- Permitir que solo el usuario administrador pueda eliminar el menú.
- Realizar una pantalla donde se pueda dar privilegios al usuario y que solo sea accedida por el administrador.
- Permitir ver en forma de preview los menús asignados a un usuario.
- Permitir dar permisos a usuarios a accesos y acciones de manera dinámica.

Proceso 10

Crear reportes, permitir generar reportes en varios formatos y poder exportar a Excel, pdf.

Requerimientos:

- El sistema debe permitir crear reportes a partir de los archivos de los analizadores.
- El sistema debe permitir generar al menos dos gráficas por pastel y barras.
- Permitir que el administrador pueda manipular las gráficas.
- El sistema debe hacer una comparación con las normas del Inen, si están fuera del rango debe indicar con un color, este color debe modificable.
- El sistema debe mostrar promedios, media, mínimo y máximo por análisis.
- El sistema debe ligar los reportes generados directamente con los clientes, para que al momento de revisar un cliente se pueda ver los reportes por requerimientos.
- El sistema debe de permitir que el Administrar mediante un clic indique que el reporte ya fue procesado.
- Realizar una pantalla que permita la búsqueda de reportes entregados a los clientes en forma cronológica.

3.2.1.1 Identificación de roles y funcionalidades.

En el laboratorio de Calidad de Leche de Cayambe de la Universidad Politécnica Salesiana, un examen inicial reveló varios posibles roles en el diseño de la aplicación web, tales como: Recepcionista, Laboratorista, Jefe de Laboratorio, Control de Calidad y Administrador del sistema.

Luego durante la especificación de casos de usos, se verificó que Control de Calidad y Jefe de Laboratorio podría unificarse en un solo rol llamado Jefe de Laboratorio.

Además se definieron tres grandes Grupos de acceso al sistema:

- Administrador, contiene los roles de Jefe de laboratorio y Administrador del sistema.
- Usuario, contiene los roles de Recepcionista y Laboratorista.

- Invitado, permite el ingreso al sistema a los clientes para ver sus resultados.

En las siguientes tablas se identifican por cada rol las tareas de la aplicación web.

Tabla 3.

Rol administrador

Rol	Tareas
Administrador Grupo Administrador	Ingresar al sistema informático
	Administrar varios tipos. Acciones: ingresar, editar, dar de baja, dar de alta, imprimir, bajar archivos, consultar.
	Administrar menús. Acciones: ingresar, editar, dar de baja, dar de alta, imprimir, bajar archivos, consultar.
	Configurar archivos aceptados por el sistema. Acciones: crear, leer, editar, consultar y dar de baja, consultar.
	Administrar usuarios. Acciones: ingresar, editar, dar de baja, dar de alta, imprimir, bajar archivos, consultar.
	Administrar perfiles Acciones: ingresar, editar, dar de baja, dar de alta, imprimir, bajar archivos, consultar.
	Configuración de pantallas Acciones: <ul style="list-style-type: none"> • Ingresar. • Editar. • Dar de baja. • Dar de alta.

Nota: Esta tabla identifica las tareas asignadas al rol Administrador

Elaborador por: Orlando Honores

Tabla 4.
Rol jefe de laboratorio

Rol	Tareas
Jefe de Laboratorio Grupo Administrador	Ingresar al Sistema
	Gestionar Clientes. <ul style="list-style-type: none"> Personas. Empresa. Productor. Laboratorista. Acciones: ingresar, editar, dar de baja, dar de alta, imprimir, subir archivos, bajar archivos, consultar.
	Gestionar Requerimientos que ofrece el laboratorio. <ul style="list-style-type: none"> Tipos de Requerimientos. Composición del Requerimiento. Acciones: ingresar, editar, dar de baja, dar de alta, imprimir, subir archivos, bajar archivos, consultar.
	Gestionar de Archivos. <ul style="list-style-type: none"> Configuración y datos Acciones: ingresar, dar de baja, dar de alta, imprimir, subir archivos, bajar archivos, consultar.
	Gestionar normas del Inen. <ul style="list-style-type: none"> Químicos y Sanitarios Acciones: ingresar, editar, dar de baja, dar de alta, imprimir, bajar archivos, consultar.
	Generar Reportes. <p>Acciones: ingresar, editar, dar de baja, dar de alta, imprimir, bajar archivos, consultar.</p>

Nota: Esta tabla identifica las tareas asignadas al rol Jefe del laboratorio
Elaborador por: Orlando Honores.

Tabla 5.
Rol Recepcionista.

Rol	Tareas
Recepcionista Grupo Usuarios	Ingresar al sistema informático
	Gestionar clientes. <ul style="list-style-type: none"> • Personas. • Empresa. • Productor. Acciones <ul style="list-style-type: none"> • Ingresar • Editar • Subir archivos • Bajar archivos • Consultar.
	Crear Requerimientos solicitados por los clientes <ul style="list-style-type: none"> • Requisición de análisis Acciones <ul style="list-style-type: none"> • Ingresar • Imprimir • Subir archivos • Bajar archivos • Consultar.
	Generador códigos de barra. <ul style="list-style-type: none"> • Lista por productor Acciones <ul style="list-style-type: none"> • Imprimir. • Consultar.

Nota: Esta tabla identifica a las tareas que corresponden al rol Recepcionista.
Elaborador por: Orlando Honores.

Tabla 6.
Rol laboratorista e invitado

Rol	Tareas
Laboratorista Grupo Usuarios	Ingresar al sistema informático
	Entrega de materiales Acciones: ingresar, imprimir, bajar archivos, consultar.
	Recibir muestras de los clientes. <ul style="list-style-type: none"> • Lotes. • Colectas. • Observaciones. Acciones: ingresar, imprimir, bajar archivos, consultar.
	Generador códigos de barra. <ul style="list-style-type: none"> • Listar. Acciones: imprimir, consultar.
Invitado Grupo Invitado	Ingresar al sistema informático
	Actualización de información adicional Acciones: ingresar, modificar y consultar.
	Consultar resultados. <ul style="list-style-type: none"> • Requerimiento. Acciones: <ul style="list-style-type: none"> • Ver • Imprimir • Bajar archivos • Consultar
	Cambiar de clave. Acciones: <ul style="list-style-type: none"> • Modificar.

Nota: Tabla que identifica las tareas asignadas al rol laboratorista.
Elaborador por: Orlando Honres.

3.2.1.2 Especificación de escenarios.

Aplicando el método OOHDM, se narrará las tareas que se ha identificado por cada rol. A continuación se describirán las tareas más comunes que tienen los roles descritos anteriormente.

Ingresar al sistema.

Los usuarios con roles de “Administrador, Jefe de Laboratorio, Recepcionista, Laboratorista e Invitado” podrán ingresar al sistema únicamente con su alias y su clave, si los datos son correctos se creará una sesión con su respectivo menú por rol asignado.

Administrar varios tipos.

Varios tipos hacen referencia a los parámetros del sistema como:

- Tipos de Identificación.
- Tipos de Requerimientos. (Nuevo o Entrega).
- Grupos de Perfiles.
- Tipos de Acciones.
- Observaciones para Colectas.
- Inen

El usuario con rol, Administrador del sistema, puede ingresar, editar, dar de baja o dar de alta a los componentes descritos en la parte superior.

Administrar menús.

El rol, Administrador del sistema, puede ingresar, editar, dar de baja o dar de alta los menús con los que cuenta el sistema. La edición o eliminación de algún menú es responsabilidad del Administrador del sistema.

Configurar archivos aceptados por el sistema.

Al momento de realizar los análisis el Laboratorio de calidad de leche de Cayambe de la Universidad Politécnica Salesiana cuenta con varios analizadores de leche rápidos y precisos como Milkoscan, Bactoscan, Fossomatic, etc. Algunos de estos

analizadores entregan los resultados en formato csv, los cuales deberían ser enviados al sistema para la obtención de resultados finales.

Es por eso que el usuario con rol Administrador del sistema, debería indicar: formato de archivos, las columnas del archivo, el índice de la columna que hace referencia al código de barras y el índice de las columnas de donde se obtendrán los resultados y por último la referencia del Inen para la comparación de los resultados.

Administrar usuarios

El usuario con rol Administrador del sistema o Jefe de laboratorio, puede crear, editar, dar de baja y dar de alta usuarios con el rol de Laboratorista, Jefe de laboratorio o Recepcionista.

El usuario Administrador del sistema o Jefe de laboratorio, debería proporcionar una clave al momento de generar el usuario.

Después de haber creado al usuario, es necesario indicar al sistema que puede acceder al mismo con su alias y la clave proporcionada.

Administrar perfiles

El usuario con rol Administrador del sistema o Jefe de laboratorio, puede crear, editar, dar de baja o dar de alta a perfiles.

Además debería indicar a qué lugares tiene acceso el perfil creado y que permisos (ingresar, editar, dar de baja, dar de alta, subir archivos, bajar archivos y consultar) podría tener.

Buscar registros (Consultar)

EL sistema ofrece una opción llamada “Buscar”, que permitirá consultar los registros según los parámetros ingresados.

Bajar registros de formato pdf. (Bajar archivos)

El usuario podrá bajar los registros mostrados por el sistema en formato pdf.

Sería factible que al momento de bajar los archivos se presente un ícono por cada formato, con el objetivo que el usuario puede identificar el tipo de archivo a bajar.

Gestionar clientes.

Los clientes del Laboratorio de Leche de Cayambe de la Universidad Politécnica Salesiana, están divididos en tres grupos persona, empresa y productores.

El usuario con el rol Jefe de laboratorio o Recepcionista al momento de seleccionar el grupo de clientes, debería ver un listado inmediato de los clientes activos de ese grupo, con el fin de poder bajar los registros o de editar un registro.

Para poder bajar los registros (conjunto de clientes activos) en forma de archivo, el usuario necesitaría saber en qué formatos podría hacerlo. Algunos formatos aceptados por el usuario son pdf, xlsx.

El usuario con el rol Jefe de laboratorio y recepcionista puede ingresar clientes, pero antes del ingreso respectivo, deberían indicar a qué grupo de cliente pertenece y disponer de un botón que indique “Nuevo Registro”.

El sistema debe validar la identificación del cliente o usuario para garantizar la integridad del registro, para esto deberá contar con un servicio que permita la obtención mediante la identificación en páginas expuestas por el gobierno para verificar los nombres o razón social

Para modificar un cliente el usuario con el rol Jefe de laboratorio y recepcionista deberá identificarlo, para esto es necesario un campo de búsqueda.

A veces para buscar un cliente con el nombre no es suficiente, ya que los clientes tienen un alias, el cual debería usarse también como parámetro de búsqueda.

Para dar de baja o de alta un registro sólo debería hacerlo el usuario con el Rol Jefe de Laboratorio.

Gestionar Requerimientos que ofrece el laboratorio.

- Tipos de Requerimientos.
- Tipos de Componentes.
- Composición del Requerimiento.

Para crear un requerimiento solicitado por el cliente, antes debería indicar al sistema que tipos de requerimientos ofrece, para lograrlo el usuario con el rol Jefe de

Laboratorio, puede crear, editar, dar de alta y dar de baja los tipos de requerimientos como “análisis, frascos y etiquetas, conservantes, etc.”, los tipos de componentes como “composición, conteo de células somáticas, etc.”, y composición de requerimientos que es una relación de los componentes con los requerimientos.

Gestionar archivos.

El usuario con el rol Jefe de laboratorio, puede subir archivos, según la configuración indicada en el escenario “Configurar archivos aceptados por el sistema”. Sin embargo a veces por complicaciones del formato del archivo no se ha podido subir el archivo, así que sería recomendable otras formas de subir el archivo, una de ellas es copiar el contenido del archivo (Copy and Paste) y que el sistema lo procese.

Gestionar normas del Inen.

El usuario con el rol Jefe de laboratorio, puede ingresar y editar las normas de Inen. (INEN Leche cruda N°0009:2012). (INEN, 2015)

Generar reportes.

El usuario con el rol Jefe de Laboratorio, puede generar reportes; para lograrlo el sistema debería unificar los resultados previamente subidos mediante el código de barras, obtener promedios, medianas, máximos, mínimos y referencias con el Inen. El usuario puede crear gráficos y enviar los resultados a dos tipos de archivos pdf y xlsx.

Crear requerimientos solicitados por los clientes.

1. Requisición de Análisis.

El usuario con el rol Recepcionista, puede crear nuevos requerimientos solicitados por los clientes.

A veces los clientes entregan archivos con información de sus vaquitas, las mismas que deben ser ingresadas al momento de hacer la requisición de análisis. El sistema debería permitir ingresar estos archivos, ya sea haciendo un upload o manualmente, ingresado vaquita por vaquita.

2. Generador Códigos de Barra.

El usuario con rol de Recepcionista o Laboratorista, puede generar imprimir códigos de barra.

3. Entrega de Materiales.

El usuario con rol Laboratorista, puede ingresar los materiales entregados a los clientes.

4. Recibir Muestras de los Clientes.

El usuario con rol Laboratorista, puede ingresar las muestras recibidas por los clientes. Para el ingreso de las muestras recibidas deberá crear un Lote, ingresar las colectas que receptan durante la mañana, e ingresar las observaciones por colectas.

3.2.1.3 Especificación de diagramas de interacción de usuario

Para cada caso de uso, se define un diagrama de interacción con el usuario (UID). UIDs representan gráficamente la interacción entre el usuario y la aplicación. Este diagrama sólo se describe el intercambio de información entre el usuario y la aplicación, sin considerar aspectos específicos de la interfaz de usuario. UML (Unified Modeling Language) no presenta similares diagrama. UML diagramas que representan la interacción, tales como la secuencia de diagramas, la colaboración diagramas, diagramas de estados, y diagramas de actividades, tenga en cuenta el intercambio de mensajes entre el objetos del sistema.

La Figura 15 muestra el UID que se ha definido para representar la interacción en la tarea de "Administrar Varios Tipos."

La Figura 16 muestra el UID que se ha definido para representar la interacción en la tarea de "Administrar Menús."

La Figura 17, muestra el UID que se ha definido para representar la interacción en la tarea de "Administrar Usuarios.". La Figura 19 muestra el UID que se ha definido para representar la interacción en la tarea de "Gestionar Clientes."

Varios tipos

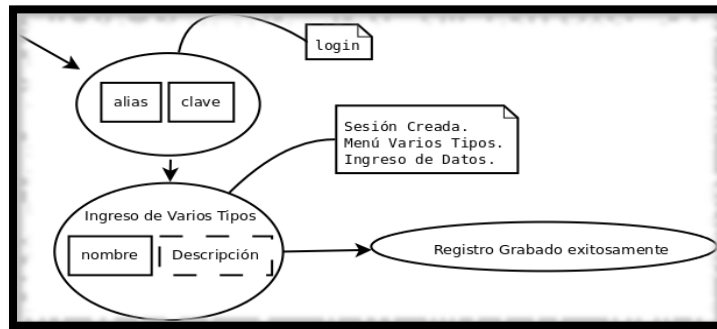


Figura 15. Administración varios tipos.
Elaborado por: Orlando Honores

Administración de menús

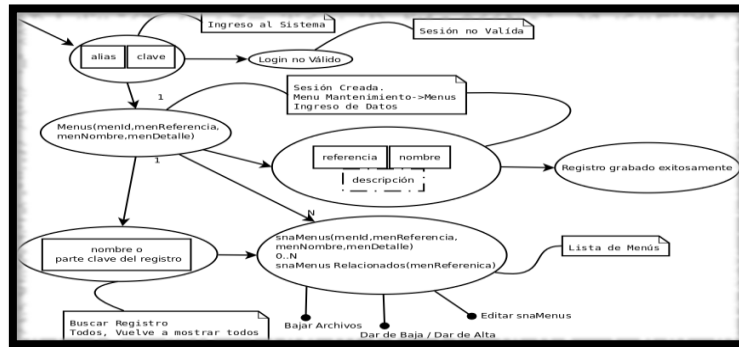


Figura 16. UID Administración de usuarios.
Elaborado por: Orlando Honores

Administrar clientes

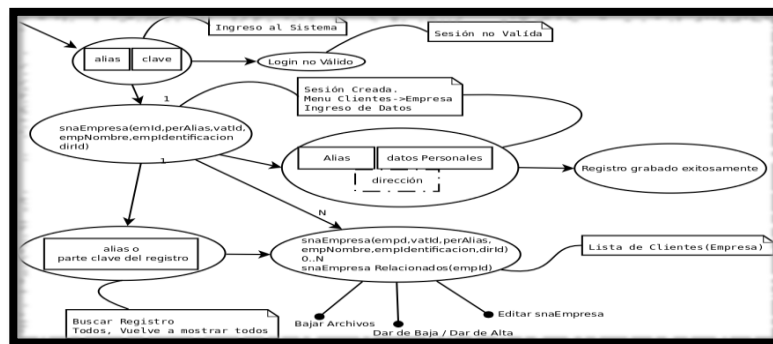


Figura 17. UID Gestión de clientes
Elaborado por: Orlando Honores

3.2.2.1 Especificación y diagramas de casos de uso

Es la agrupación de escenarios en unidades funcionales, de las tareas y roles analizados en la Especificación de Escenarios. Para llevar a cabo esta tarea se utiliza el lenguaje de modelamiento unificado, en particular el Diagrama de Caso de Uso, que nos ayudará a ver en forma gráfica la función de los roles, mas sin embargo se debe especificar las acciones correspondientes en los Casos de uso.

A continuación se presentarán en forma de tablas las especificaciones de los casos de uso, seguidos por su respectivo diagrama.

Tabla 7.

Casos de uso administrar varios tipos.

Nombre		Administrar Varios Tipos.
Descripción: permite gestionar los diferentes tipos de objetos que se utilizan en el sistema. Estos objetos son: <ul style="list-style-type: none">• Tipos de identificación.• Tipos de requerimientos. (Nuevo o Entrega).• Grupos de perfiles.• Tipos de acciones.• Observaciones para colectas.• Inen		
Actores <ul style="list-style-type: none">• Administrador del sistema		
Precondiciones <ul style="list-style-type: none">• El usuario debe estar logueado.		
Flujo normal: <ul style="list-style-type: none">• El usuario accede al Rol de Administrador una vez ingresado su alias y su clave.• El sistema valida los datos, aplicando PASSPORT, para determinar si se trata de un usuario con grupo Administrador, para luego ver qué permisos fueron asignados al usuario.• El usuario puede navegar por todo el menú “Mantenimiento” sin ninguna restricción.• El usuario puede escoger cualquier submenú descrito anteriormente.• El usuario puede ingresar, editar, ver, dar de baja, dar de alta, bajar archivos en formato xls, pdf; además puede realizar consultas. El usuario puede ver la lista de todos los tipos ingresados en el sistema.• El usuario Administrador es responsable de la información que formará parte para el resto de actores del sistema.		

Flujo Alternativo:

- Validación de datos.
- Valida el nombre del tipo que sea único en el sistema.

Nota: Se describe la especificación de casos de uso para la gestión de varios tipos
Elaborado por: Orlando Honores.

En la figura 19, se detalla el caso de uso Administrar varios tipos

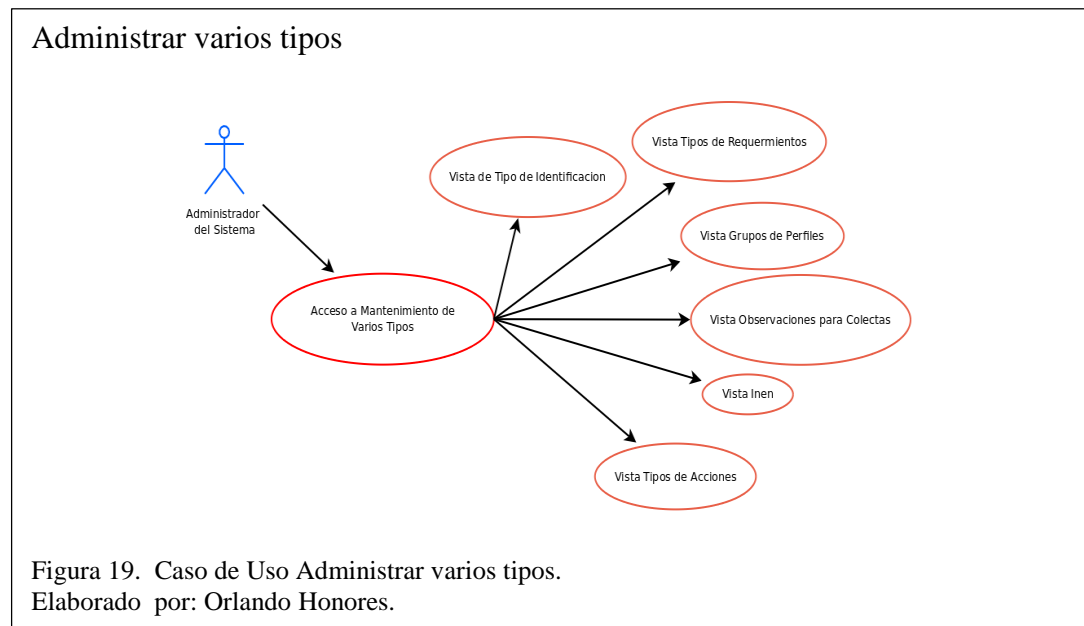


Tabla 8.
Caso de uso administrar de menú

Nombre:	Administrar de menú.
<p>Descripción: permite gestionar el menú del sistema, que consiste en:</p> <ul style="list-style-type: none"> • Referencia o Menú principal. • Nombre • Descripción. • Url (Este campo es llenado automáticamente al presionar “Generar”, una vez que se ha ingresado los campos ya descritos.) • Tabla a la que hará referencia el menú • Referencia a un archivo, para archivos estáticos. 	

<p>Actores</p> <ul style="list-style-type: none"> Administrador del sistema
<p>Precondiciones</p> <ul style="list-style-type: none"> El usuario debe estar logueado.
<p>Flujo normal:</p> <ul style="list-style-type: none"> El usuario accede al Rol de Administrador una vez ingresado su alias y su clave. El sistema valida los datos, aplicando PASSPORT, para determinar si se trata de un usuario con grupo Administrador, para luego ver qué permisos fueron asignados al usuario. El usuario puede navegar por todo el menú “Mantenimiento->Menús” sin ninguna restricción. El usuario puede ingresar, editar, ver, dar de baja, dar de alta, bajar archivos en formato xls, pdf; además puede realizar consultas. El usuario puede ver la lista de todos los menús con los que cuenta el sistema. El usuario Administrador es responsable de la información que formará parte para el resto de actores del sistema. El usuario podrá incluir archivos externos del sistema, los cuales formarán parte de un nuevo sub-menú o ítem. Cada menú hace referencia a una tabla de la base de datos, la cual es configurada previamente.
<p>Flujo alternativo:</p> <ul style="list-style-type: none"> Validación de datos. Valida el nombre del menú que sea único en el sistema.

Nota: Se describe la especificación de casos de uso para la gestión de Menús
Elaborado por: Orlando Honores.

En la figura 20, se detalla el caso de uso administrar menús

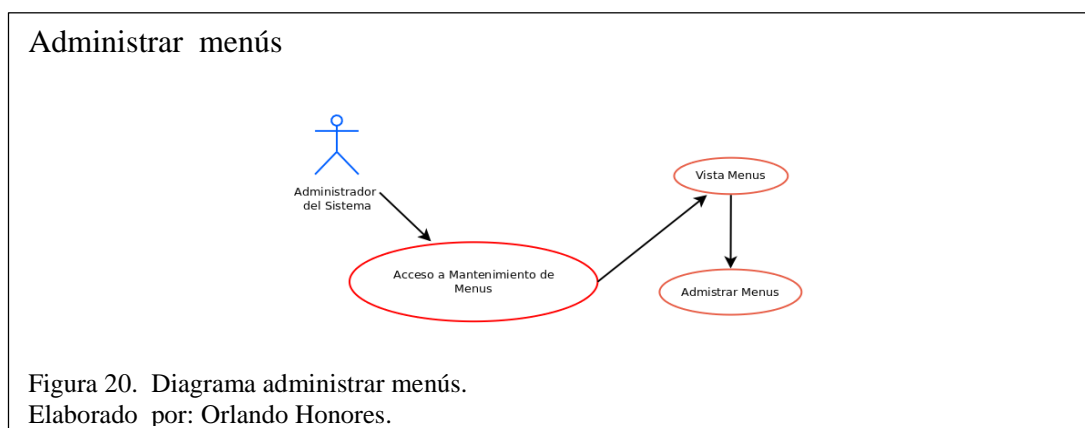


Tabla 9.
Casos de uso administrar usuarios.

Nombre:	Administrar usuarios.
Descripción: permite gestionar los usuarios que pueden acceder al sistema. Esto involucra crear, editar, dar de baja y dar de alta.	
Actores <ul style="list-style-type: none"> Administrador del sistema 	
Precondiciones <ul style="list-style-type: none"> El usuario debe estar logueado. 	
Flujo Normal: <ul style="list-style-type: none"> El usuario accede al Rol de Administrador una vez ingresado su alias y su clave. El sistema valida los datos, aplicando PASSPORT, para determinar si se trata de un usuario con grupo Administrador, para luego ver qué permisos fueron asignados al usuario. El usuario puede navegar por todo el menú “Mantenimiento->Usuarios” sin ninguna restricción. El usuario puede ingresar, editar, ver, dar de baja, dar de alta, bajar archivos en formato xls, pdf; además puede hacer una búsqueda inteligente, solo debe ingresar parte clave del usuario a buscar. El usuario puede ver la lista de todos los usuarios activos del sistema. El usuario Administrador puede activar o desactivar el ingreso al sistema a Clientes que se los ha ingresado como “Persona, Empresa, Productor e Invitado” El usuario Administrador puede cambiar de rol a cualquier usuario. El usuario Administrador es responsable de la información que formará parte para el resto de actores del sistema. 	
Flujo alternativo: <ul style="list-style-type: none"> Validación de datos. Valida el nombre del usuario que sea único en el sistema. 	

Nota: Se describe la especificación de casos de uso para la gestión de usuarios
Elaborado por: Orlando Honores.

Tabla 10.

Casos de uso administrar la configuración de archivos aceptados por el sistema.

Nombre:	Administrar la configuración de archivos aceptados por el sistema.
<p>Descripción: permite gestionar los archivos que ser receptará y el contenido del mismo:</p> <ul style="list-style-type: none"> • Archivos de resultados de análisis. • Configuración de las columnas si se trata de un archivo de texto, csv o Excel. • Archivos de nuevos ítems (vaquitas) para la creación de nuevos requerimientos. A estos archivos se los conoce como “Archivos de Clientes” 	
<p>Actores</p> <ul style="list-style-type: none"> • Administrador del sistema 	
<p>Precondiciones</p> <ul style="list-style-type: none"> • El usuario debe estar logueado. 	
<p>Flujo normal:</p> <ul style="list-style-type: none"> • El usuario accede al Rol de Administrador una vez ingresado su alias y su clave. • El sistema valida los datos, aplicando PASSPORT, para determinar si se trata de un usuario con grupo Administrador, para luego ver qué permisos fueron asignados al usuario. • El usuario puede navegar por todo el menú “Mantenimiento->Archivos” sin ninguna restricción. • El usuario puede ingresar, editar, ver, dar de baja, dar de alta, subir archivos, bajar archivos según el formato; además puede realizar consultas. • El usuario puede ver la lista de todos los archivos de configuración con los que cuenta el sistema. • El usuario Administrador es responsable de la información que formará parte para el resto de actores del sistema. • El usuario podrá incluir archivos externos del sistema, los cuales formarán parte de un nuevo sub-menú o ítem. 	
<p>Flujo alternativo:</p> <ul style="list-style-type: none"> • Validación de datos. • Valida el nombre del archivo que sea único en el sistema. 	

Nota: Se describe la especificación de casos de uso para la gestión de archivos
Elaborado por: Orlando Honores.

En la figura 21, se detalla el caso de uso Administrar configuración de archivos aceptados por el sistema

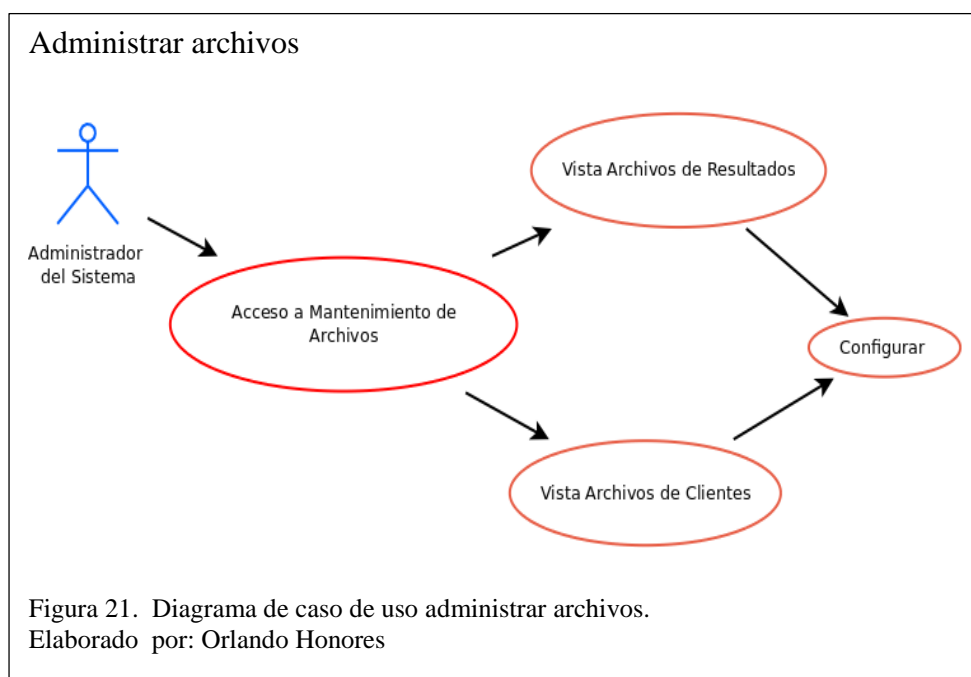


Tabla 11.
Casos de uso Administrar Perfiles.

Nombre:	Administrar perfiles.
Descripción: permite gestionar los roles del sistema.	
Cada rol está compuesto por un conjunto acciones y accesos al sistema.	
Actores <ul style="list-style-type: none"> Administrador del sistema 	
Precondiciones <ul style="list-style-type: none"> El usuario debe estar logueado. 	
Flujo normal: <ul style="list-style-type: none"> El usuario accede al Rol de Administrador una vez ingresado su alias y su clave. El sistema valida los datos, aplicando PASSPORT, para determinar si se trata de un 	

<p>usuario con grupo Administrador, para luego ver que permisos fueron asignados al usuario.</p> <ul style="list-style-type: none"> • El usuario puede navegar por todo el menú “Mantenimiento->Varios Tipos” sin ninguna restricción. • El usuario puede ingresar, editar, ver, dar de baja, dar de alta, bajar resultados en formato xls, pdf. Además puede realizar consultas • El usuario debe seleccionar Grupos de Perfiles • El usuario puede ver la lista de perfiles con los que cuenta el sistema. • Se debe seleccionar el perfil a gestionar. • En una nueva pantalla el usuario podrá ver los menús que corresponden al perfil. • El usuario podrá añadir o eliminar menús o submenús del perfil. • El usuario podrá añadir o eliminar acciones al recurso del submenú. • El usuario podrá ver una vista previa de cómo va como se verá el perfil. • El usuario Administrador es responsable de la información que formará parte para el resto de actores del sistema.
<p>Flujo alternativo:</p> <ul style="list-style-type: none"> • Validación de datos. • Valida el nombre del perfil que sea único en el sistema.

Nota: Se describe la especificación de casos de uso para la gestión de perfiles
Elaborado por: Orlando Honores.

Tabla 12.
Caso de uso Gestionar clientes

Nombre:	Gestionar clientes.
<p>Descripción: permite gestionar los clientes, los cuales están divididos en tres grupos</p> <ul style="list-style-type: none"> • Personas • Empresas • Productores 	
<p>Actores</p> <ul style="list-style-type: none"> • Jefe de laboratorio • Recepcionista. 	
<p>Precondiciones</p> <ul style="list-style-type: none"> • El usuario debe estar logueado. 	

Flujo normal:

- El usuario accede ingresando su alias y contraseña.
- El “Jefe de Laboratorio” o “Recepcionista” escoge Persona, Empresa, Productor o Invitado en la Pantalla Clientes.
- El sistema muestra la lista de clientes según el tipo de cliente escogido.
- El “Jefe de Laboratorio” o “Recepcionista” puede crear un archivo (formato xls o pdf) de los clientes mostrados por el sistema.
- El “Jefe de Laboratorio” o “Recepcionista” puede realizar una búsqueda de clientes tipo “autocomplete”, para esto debe ingresar parte del nombre, identificación o ubicación del cliente.
- El sistema desplegará una lista de los registros encontrados.
- El “Jefe de Laboratorio” o “Recepcionista” selecciona un registro de la lista “autocomplete”.
- El “Jefe de Laboratorio” o “Recepcionista” agrega un registro, haciendo click en el botón “NUEVO REGISTRO”.
- El sistema muestra un formulario de ingreso de datos, para el nuevo registro.
- El “Jefe de Laboratorio” o “Recepcionista” se hace responsable de la información entregada al sistema.
- El “Jefe de Laboratorio” o “Recepcionista” puede editar un registro.
- El “Jefe de Laboratorio” puede dar de baja a un registro.
- El “Jefe de Laboratorio” puede dar de alta a un registro, siempre y cuando el registro esté dado de baja.

Flujo alternativo:

- Validación de datos.
- Valida el nombre del perfil que sea único en el Sistema.
- Valida los nombres del cliente que sea único.
- Valida la identificación de los clientes sea única.

Nota: Se describe la especificación de casos de uso para la gestión de Clientes
Elaborado por: Orlando Honores.

Tabla 13.

Caso de uso Gestionar requerimientos que ofrece el Laboratorio

Nombre:	Gestionar Requerimientos que ofrece el laboratorio.
Descripción: gestionar los diferentes análisis que ofrece el Laboratorio de calidad leche de Cayambe.	

<p>Actores:</p> <ul style="list-style-type: none"> • Jefe de laboratorio
<p>Precondiciones:</p> <ul style="list-style-type: none"> • El usuario debe estar logueado.
<p>Flujo normal:</p> <ul style="list-style-type: none"> • El usuario accede al Rol de Administrador una vez ingresado su alias y su clave. • El sistema valida los datos, aplicando PASSPORT, para determinar si se trata de un usuario con grupo Administrador, para luego ver que permisos fueron asignados al usuario. • El usuario puede navegar por todo el menú “Mantenimiento->Varios Tipos”, “Mantenimiento->Tipos de Componentes” “Mantenimiento->Composición de Requerimiento”, sin ninguna restricción. • El usuario selecciona Varios Tipos. • El usuario ve una Lista de todos los registros debe hacer una búsqueda según la referencia del requerimiento • El usuario selecciona Tipos de Componentes. • El usuario ve una Lista de todos los componentes. • El usuario provee información para el ingreso de un nuevo tipo de componente. • El usuario puede modificar el registro, escogiéndolo de la lista de componentes. • El usuario puede eliminar el registro, escogiéndolo de la lista de componentes. • El usuario selecciona Composición de Requerimientos. • El usuario relaciona los componentes con los tipos de requerimientos • El usuario puede ingresar, editar, ver, dar de baja, dar de alta, bajar archivos en formato xlsx, pdf; además puede hacer una búsqueda inteligente, solo debe ingresar parte clave del archivo a buscar. • El usuario Jefe de Laboratorio es responsable de la información que formará parte para el resto de actores del sistema.
<p>Flujo alternativo:</p> <ul style="list-style-type: none"> • Validación de datos. • Valida el nombre del tipo de requerimiento que sea único. • Valida el nombre del tipo de composición que sea único.

Nota: Se describe la especificación de casos de uso para la gestión de requerimientos de análisis
Elaborado por: Orlando Honores.

Tabla 14.

Casos de uso recibimiento de muestras de leche para el análisis

Nombre:	Recibir muestras de leche de las vacas de los diferentes productores que tiene un cliente.
Descripción: permite ingresar las muestras recibida de los clientes	
Actores: <ul style="list-style-type: none"> • Laboratorista 	
Precondiciones: <ul style="list-style-type: none"> • El usuario debe estar logueado. 	
Flujo normal: <ul style="list-style-type: none"> • El usuario accede al Rol de Administrador una vez ingresado su alias y su clave. • El sistema valida los datos, aplicando PASSPORT, para determinar si se trata de un usuario con grupo Administrador, para luego ver que permisos fueron asignados al usuario. • El usuario crea un nuevo lote. • El usuario lee el código de barras de la muestra para identificar el cliente y requerimiento. • El sistema muestra el detalle del requerimiento, es decir los análisis solicitados, los materiales entregados y el saldo de materiales por entregar. • El usuario ingresa el contacto o persona encargada de entregar las muestras • El usuario ingresa la fecha de la colecta. • El usuario ingresa la temperatura. • El usuario ingresa la cantidad total de muestras recibidas. • El sistema muestra una nueva pantalla para que el usuario confirme los análisis que se elaborarán. • El usuario ingresa las observaciones por colectas. • El usuario graba la información. • El sistema muestra la información del lote creado. • Esta información consta de un historial de colectas recibidas, historial de análisis confirmados e historial de observaciones por colectas. • El usuario con rol Laboratorista es responsable de la información que formará parte para el resto de actores del sistema. 	
Flujo alternativo: <ul style="list-style-type: none"> • Validación de Datos. 	

- Valida que el nuevo lote empiece por los dos últimos dígitos del presente año.
- Valida que la cantidad muestras recibidas no sea mayor que la del material entregado por requerimiento.
- Valida que la cantidad de confirmación de análisis no sea mayor que la solicitada.

Nota: Se describe la especificación de casos de uso para la gestión de muestras de leche para los análisis que ofrece el laboratorio.
Elaborado por: Orlando Honores.

Tabla 15.
Casos de uso Gestionar reportes.

Nombre:	Generar reportes.
Descripción: permite gestionar los reportes.	
Actores: <ul style="list-style-type: none"> • Jefe de Laboratorio. 	
Precondiciones: <ul style="list-style-type: none"> • El usuario debe estar logueado. 	
Flujo Normal: <ul style="list-style-type: none"> • El usuario accede al Rol de Administrador una vez ingresado su alias y su clave. • El sistema valida los datos, aplicando PASSPOT, para determinar si se trata de un usuario con grupo Administrador, para luego ver que permisos fueron asignados al usuario. • El usuario puede navegar por todo el menú “Reportes->Ver” sin ninguna restricción. • El usuario accede a los resultados de los análisis en forma de pdf, xls y gráficamente. 	
Flujo alternativo: <ul style="list-style-type: none"> • Validación de datos. • Valida el alias y la clave para acceder al sistema 	

Nota: Se describe la especificación de casos de uso para la gestión de reportes.
Elaborado por: Orlando Honores.

Tabla 16.
Casos de uso gestionar normas del Inen.

Nombre:	Gestionar el almacenamiento de las normas del Inen.
Descripción: permite ingresar los rangos que deberían estar los análisis, según la norma del Inen.	
Actores: <ul style="list-style-type: none"> • Jefe de Laboratorio 	
Precondiciones: <ul style="list-style-type: none"> • El usuario debe estar logueado. 	
Flujo Normal: <ul style="list-style-type: none"> • El usuario accede al Jefe de Laboratorio una vez ingresado su alias y su clave. • El sistema valida los datos, aplicando PASSPORT, para determinar si se trata de un usuario con grupo Administrador, para luego ver que permisos fueron asignados al usuario. • El usuario puede navegar por todo el menú “Mantenimiento->Inen” sin ninguna restricción. • El usuario escoge Químicos o Sanitarios para el ingreso de los rangos que indica el Inen. • El usuario modifica • El usuario da de baja. • El usuario da de alta. • El usuario consulta los registros. • El usuario puede bajar los registros en formato xls o pdf. 	
Flujo Alternativo: <ul style="list-style-type: none"> • Validación de datos. • Valida la descripción que sea único en el sistema. 	

Nota: Se describe la especificación de casos de uso para la gestión el almacenamiento de las normas del Inen.

Elaborado por: Orlando Honores.

Tabla 17.

Caso de Uso para gestionar la configuración de pantallas que tiene el sistema

Nombre:	Gestionar pantallas.
Descripción: permite la configuración de pantallas según la tabla elegida.	
Actores: <ul style="list-style-type: none"> Administrador 	
Precondiciones: <ul style="list-style-type: none"> El usuario debe estar logueado. 	
Flujo normal: <ul style="list-style-type: none"> El usuario accede al Rol de Administrador una vez ingresado su alias y su clave. El sistema valida los datos, aplicando PASSPORT, para determinar si se trata de un usuario con grupo Administrador, para luego ver que permisos fueron asignados al usuario. El usuario puede navegar por todo el menú “Mantenimiento->Configuración” sin ninguna restricción. El usuario identifica el esquema de la tabla a convertir en pantalla El usuario aplica validaciones según el tipo de dato El usuario activa o inactiva que datos se mostrarán en la pantalla. El usuario valida los campos de referencia a otras tablas según los servicios proporcionados. El usuario puede ver un preview de cómo quedará la pantalla El usuario puede hacer uso de servicios preestablecidos como la validación de identificación, etc. 	
Flujo alternativo: <ul style="list-style-type: none"> Validación de datos. Valida la descripción que sea único en el sistema. 	

Nota: Se describe la especificación de casos de uso para la gestión de pantallas.

Elaborado por: Orlando Honores.

La figura 22, describe el diagrama de caso de uso: Gestionar requerimiento, que hace referencia a los análisis que ofrece el sistema.

Requerimientos que ofrece el Laboratorio

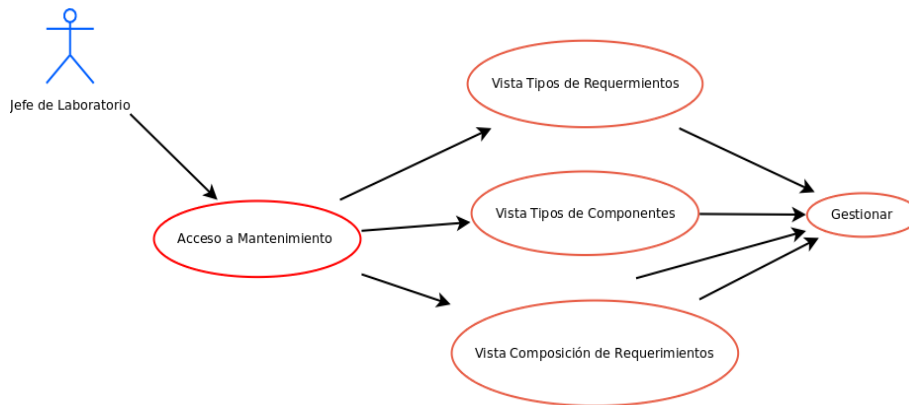


Figura 22. Diagrama caso de uso requerimientos.
Elaborado por: Orlando Honores.

La figura 23, describe el diagrama de caso de uso: Recibir muestras de leche para el análisis.

Recibimiento de muestras de leche para el análisis.

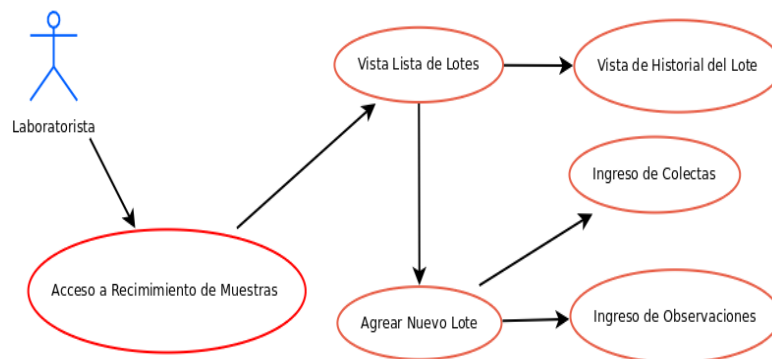
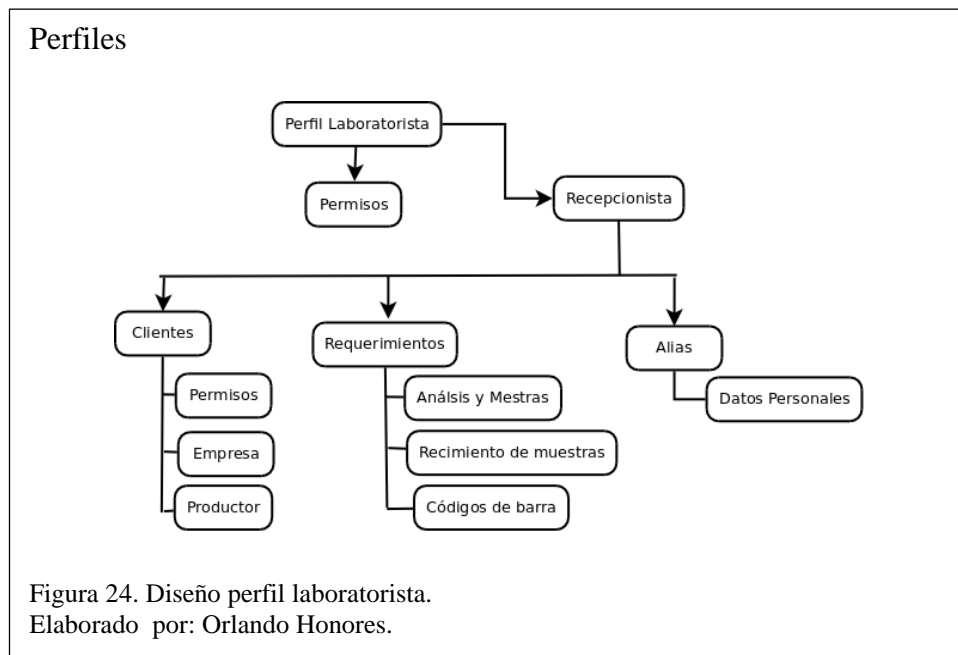


Figura 23. Diagrama caso de uso de recibimiento de muestras de leche para el análisis.
Elaborado por: Orlando Honores.

3.2.3 Diseño navegacional



La capa navegacional se compone de objetos construidos a partir de objetos conceptuales, y constituyen en general los elementos canónicos de las aplicaciones hipertexto tradicionales: nodos, enlaces, anclas y estructuras de acceso. Sin embargo, estas clases pueden extender el comportamiento característico para funcionar como adaptadores de los objetos conceptuales y delegar así operaciones específicas del dominio. Clase navegacional de Perfiles, representa la navegación de usuarios con un determinado perfil, ver figura 24.

3.2.3.1 Esquema de contexto navegacional

La imagen 25, representa el esquema navegacional del sistema informático. Cada nodo representa una vista del registro. La navegación del sistema se compone de una página principal en donde los usuarios poder hacer su login, según el rol de cada usuario se presenta otra página que estará constituida del menú y acciones asignado al rol que pertenece el usuario.

Diseño navegacional

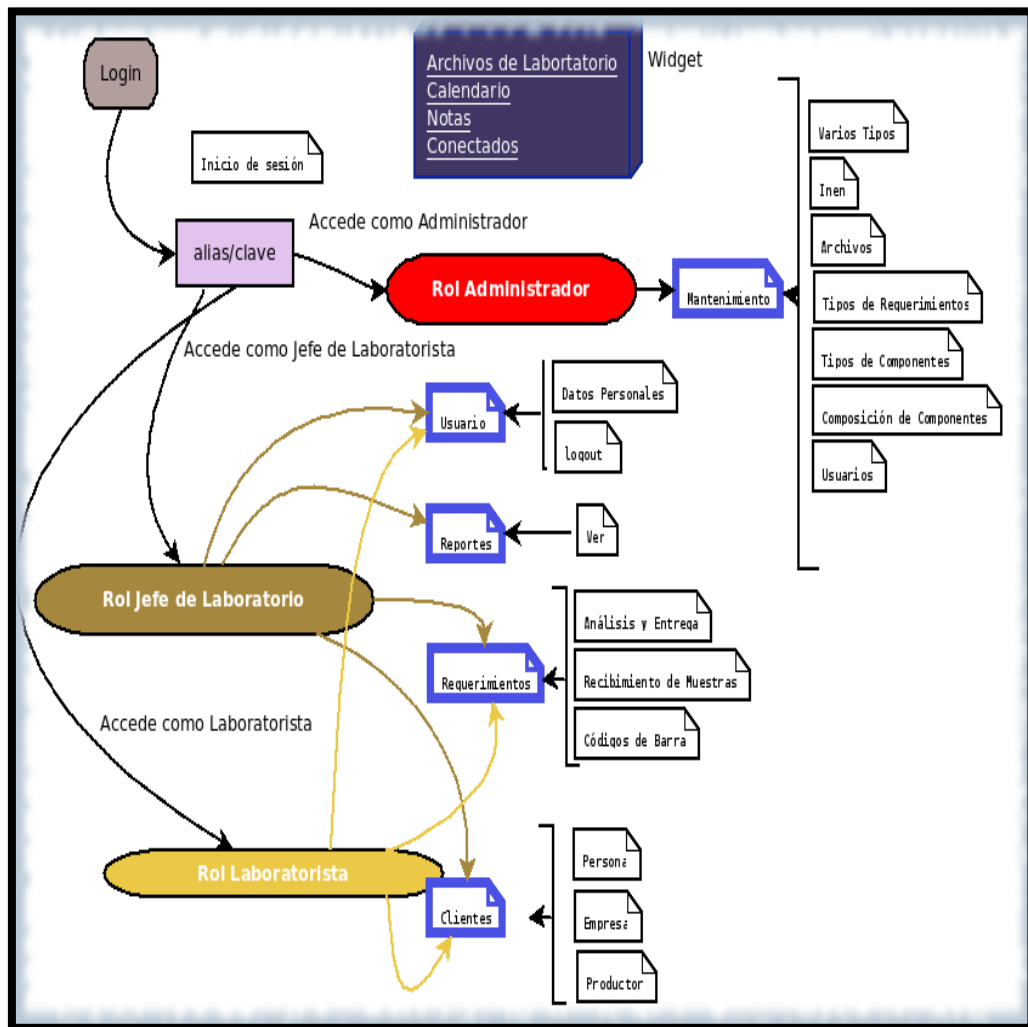


Figura 25. Diseño navegacional.
Elaborado por: Orlando Honores

3.2.4 Diseño de interfaz abstracta

La función principal de ese diseño es la creación de plantillas que van estar relacionadas con la pantalla final, que el usuario visualizará al momento de interactuar con el sistema informático. Estas interfaces tienen que ver directamente con los escenarios y casos de usos antes mencionados. Los roles y tareas son parte fundamental para la creación de estas pantallas. Se debe especificar qué objetos de interfaz el usuario percibe, hacer una distinción entre las operaciones de navegación y las operaciones de la interfaz.

Se describen dos Plantillas, la primera que es llamada por todas las interfaces, y la segunda es utilizada al ingresar por primera vez a la aplicación web, es decir representa la pantalla de inicio de la aplicación.

Plantilla principal. Es está compuesto por 3 plantillas:

Menú superior e inferior (cabecera y pie de página)

Contiene el ADV “MenuDinámico”, que se instancia una vez que el usuario se ha logueado al sistema. Se define un menú por Perfil.

Contenido

Contiene un ADV “ListarRegistros” el cual es utilizado para mostrar:

- Título de la Entidad a mostrar
- Campo de Búsqueda.
- Botones ingreso de datos y buscar registros.
- Lista de registros, muestra los registros en una tabla por cada registro se activa un ADV link “Registro_Form”.
- Es utilizado en todas las pantallas de inicio.

Código extra, se define los diálogos que se utilizarán para la inserción, edición y eliminación de registros. Estos serán llamados por los distintos componentes que están ubicados en el Layout Contenido.

Componente ADV “Registro_Form” el cual es utilizado para mostrar:

- Título de la Entidad a mostrar
- Campos (Input TextFields) de la Entidad
- Botón Aceptar.
- Botón Cancelar.

Es utilizado para ingreso y modificación de registros de registros. Los componentes ListarRegistros y Registro_Form, se encargan llaman a este componente.

A continuación se muestran imágenes que representan las plantillas y ADV’s de la aplicación:

Página principal

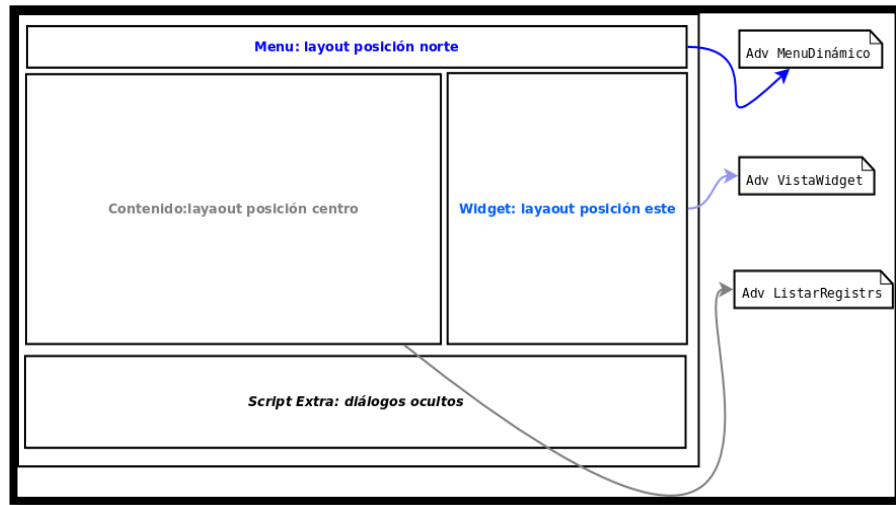


Figura 26. Plantilla de la página principal.
Elaborado por: Orlando Honores

La figura 26, se muestra una imagen que hace referencia al layout principal, que consta de un menú en la parte superior, un cuerpo en el centro y un pie de página en la parte inferior.

Listar registros.

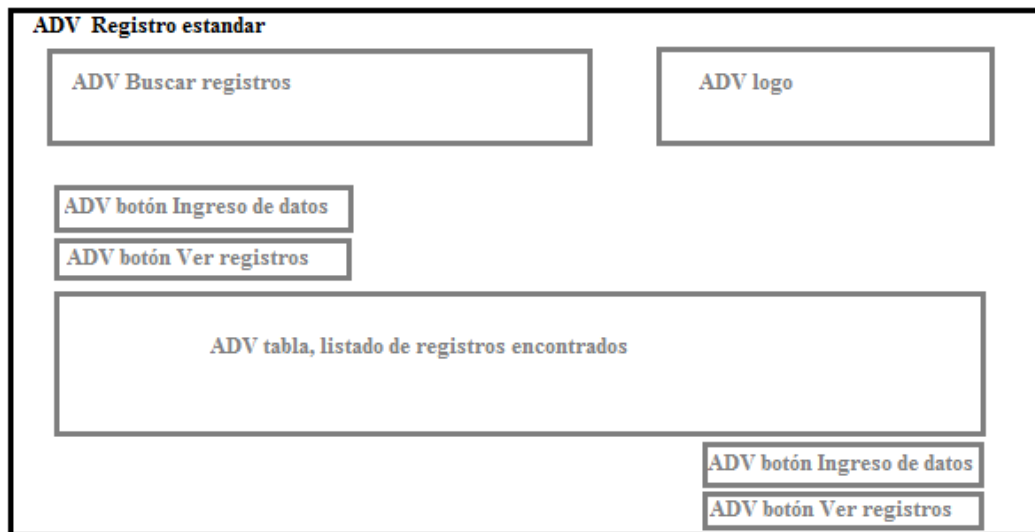


Figura 27. Adv. Listar registrar, para ver los registros del sistema.
Elaborado por: Orlando Honores.

La figura 27, se muestra una imagen ADV llamada “Registro Estándar”, como se puede observar cuenta con una sección de búsqueda en la parte superior, con un logo en la parte superior derecha, dos botones el uno para el ingreso de datos y el otro para ver los registros. En la mitad existe un ADV tabla e donde se mostraran todos los registros.

La figura 28, representa un ADV representa el ingreso de datos de un cliente.

Ingreso de datos.

The diagram shows an ADV form titled "ADV DialogoRegistro "Empresa"". The form is divided into several sections:

- Top Section:**
 - Nuevo Registro:** Static TextField
 - RUC:** OutputTextField with value `# {snaEmpresa.selected.empRuc}; OutputTextField`
 - Empresa:** OutputTextField with value `# {snaEmpresa.selected.rucEmpresa}; OutputTextField`
 - ALIAS:** OutputTextField with value `# {snaEmpresa.selected.perid.perAlias}; OutputTextField`
- Direction Section:**
 - Dirección:** Static TextField
 - Ubicación:** OutputTextField with value `# {snaEmpresaController.selected.dirId.pagGeonameId.pagName}; OutputTextField`
 - Dirección:** OutputTextField with value `# {snaEmpresaController.selected.dirId.dirDireccion}; OutputTextField`
 - Teléfono:** OutputTextField with value `# {snaEmpresaController.selected.dirId.dirTelefono}; OutputTextField`
 - Email:** OutputTextField with value `# {snaEmpresaController.selected.dirId.dirEmail}; OutputTextField`
- Bottom Section:**
 - Aceptar:** button
 - Cancelar:** button

On the right side of the form, there are two database schema diagrams:

sna_empresa:

emp_id	inst	Nullable = false
emp_id	inst	Nullable = true
emp_id	inst	Nullable = false
emp_id	inst	Nullable = false
emp_id	inst	Nullable = true
emp_id	inst	Nullable = true

sna_empresa_dir:

dir_id	inst	Nullable = false
dir_id	inst	Nullable = false
dir_id	inst	Nullable = true
dir_id	inst	Nullable = true
dir_id	inst	Nullable = true

Figura 28. Adv. Edición de datos, para la modificación de registros
Elaborado por: Orlando Honores.

3.2.4.1 Diagrama relacional

Diagrama relacional

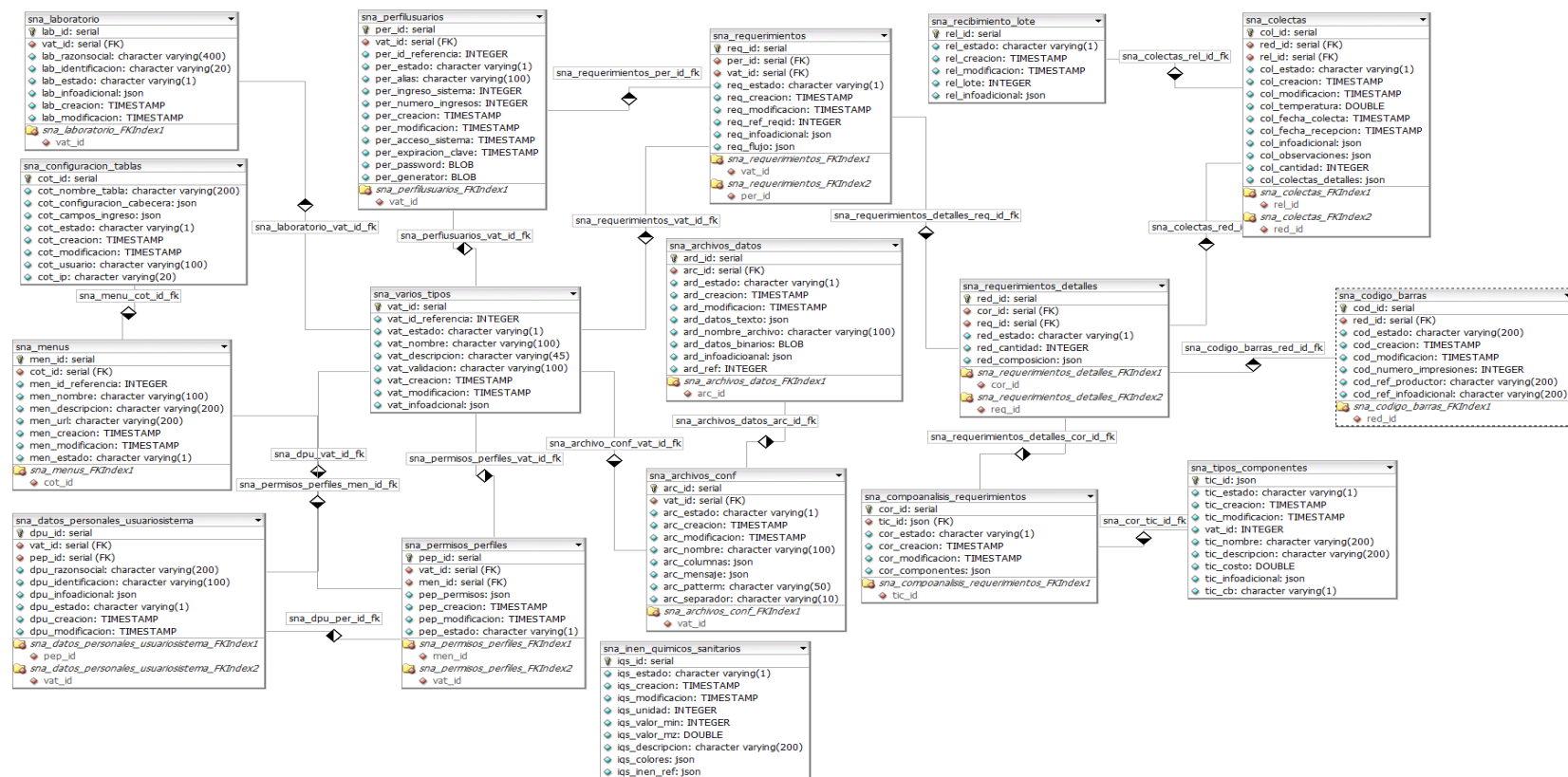
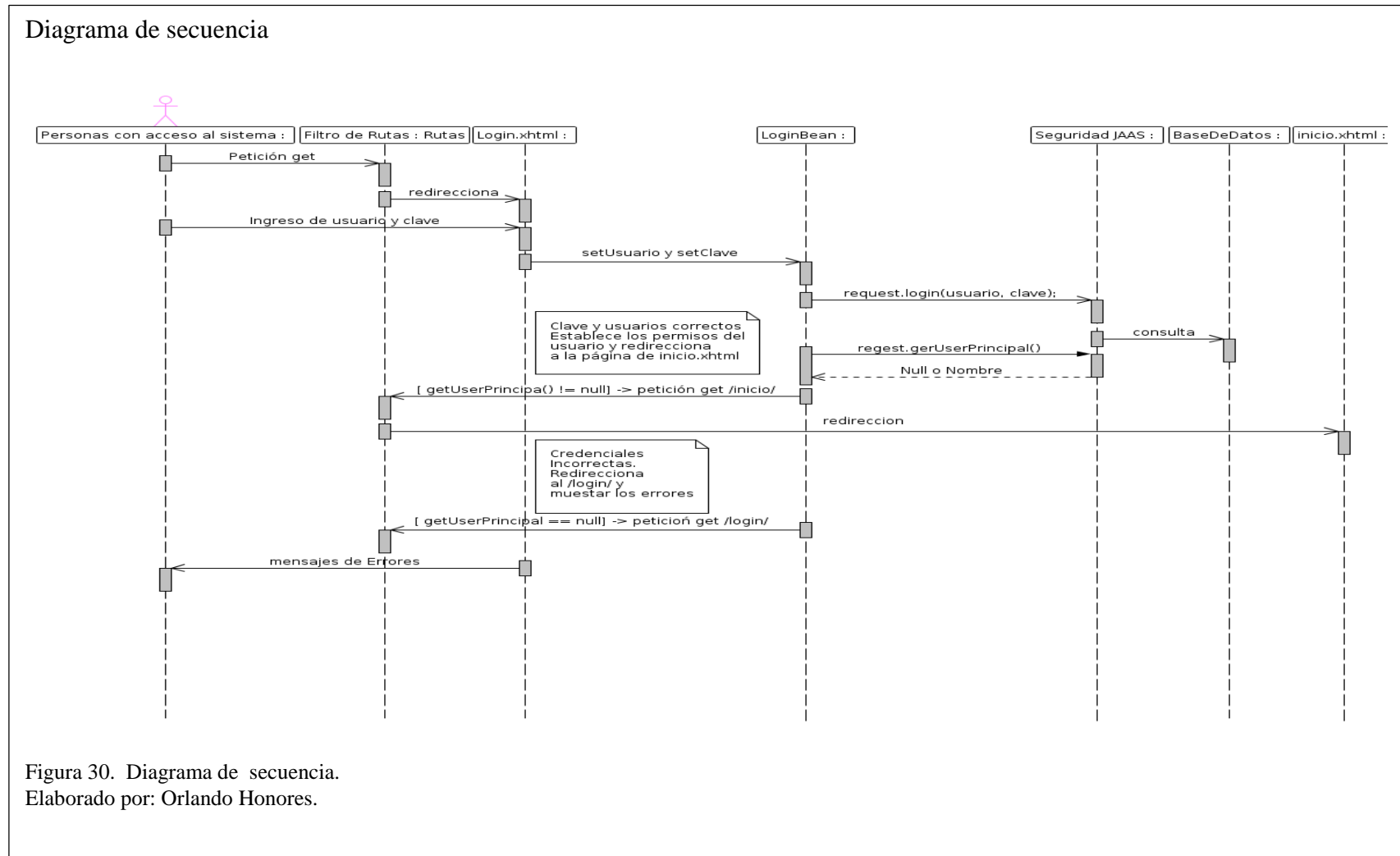


Figura 29. Diagrama modelo relacional de la base de datos en Postgres.
Elaborado por: Orlando Honores

3.2.4.2 Diagrama de secuencia



CAPÍTULO 4

DESARROLLO DEL SISTEMA PROPUESTO

4.1 Módulos de la aplicación

El desarrollo del software se encuentra organizado en módulos, que ya fueron descritos en la fase de diseño:

- Clientes.
- Requerimientos.
- Recepción de materiales.
- Análisis de materiales y entrega de muestras.

En la figura 31, se observa los diferentes módulos, en forma gráfica, que conforma el software desarrollado para procesar los resultados obtenidos sobre la calidad de leche de las muestras receptadas en el Laboratorio

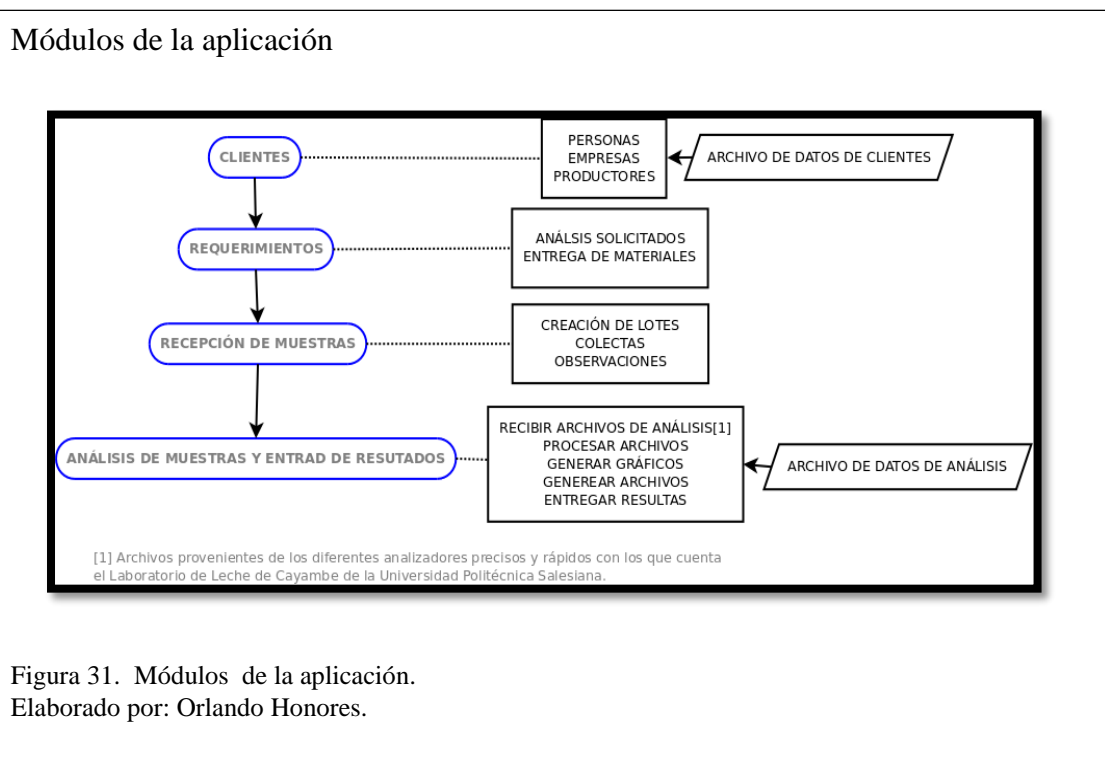


Figura 31. Módulos de la aplicación.
Elaborado por: Orlando Honores.

Diagrama de Clases de clientes.

La figura 32, representa en forma gráfica la relación entre las clases, que hacen referencia al módulo clientes.

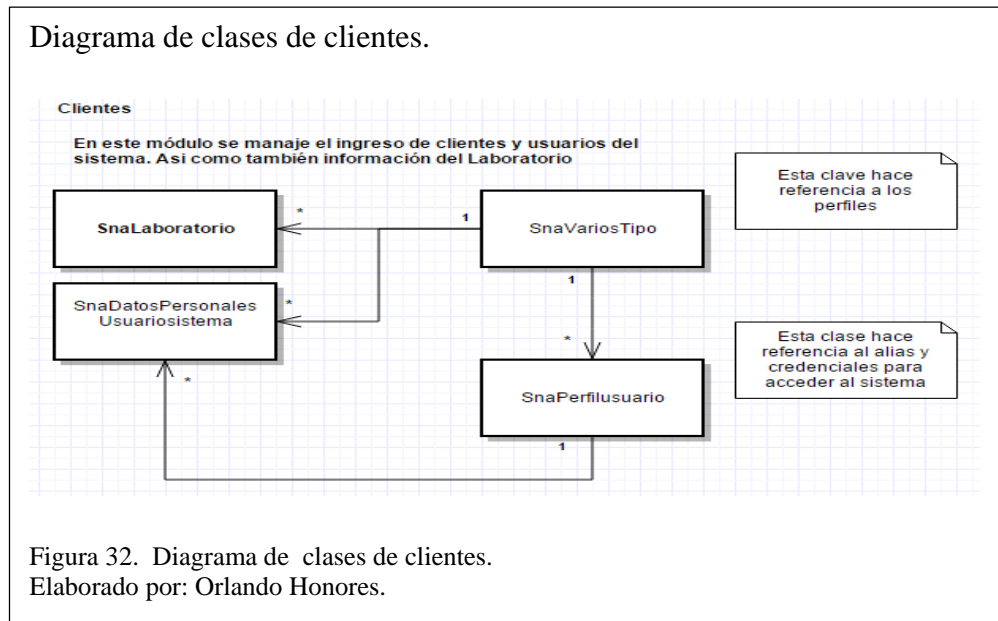


Figura 32. Diagrama de clases de clientes.
Elaborado por: Orlando Honores.

Diagrama de Clases de requerimientos.

La figura 33, representa en forma gráfica la relación entre las clases, que hacen referencia al módulo requerimientos.

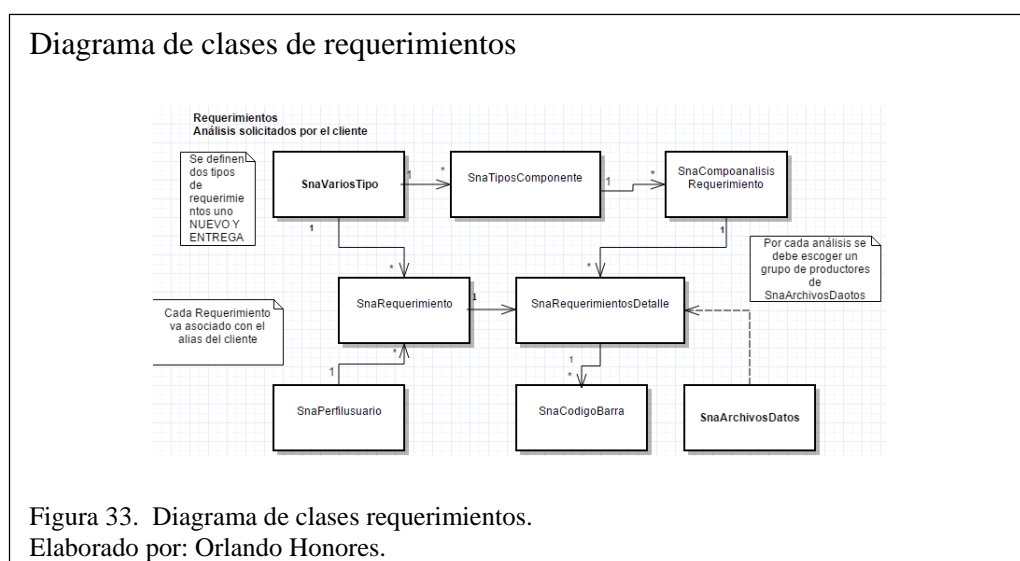


Figura 33. Diagrama de clases requerimientos.
Elaborado por: Orlando Honores.

Diagrama de Clases de Recepción de Muestras.

La figura 34 representa en forma gráfica la relación entre las clases, que hacen referencia al módulo de muestras.

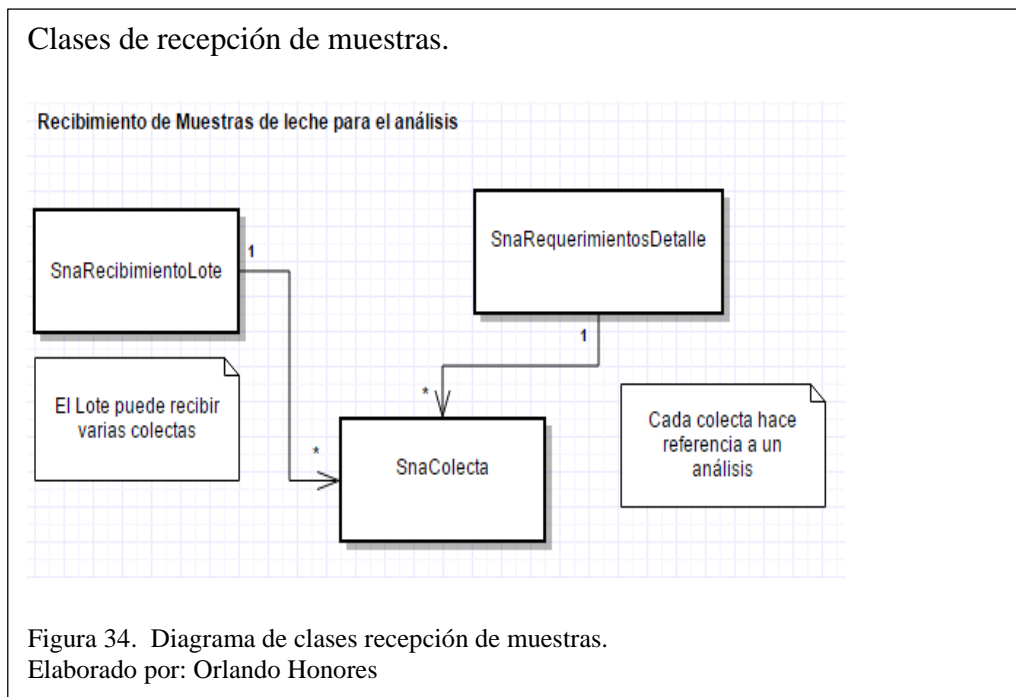
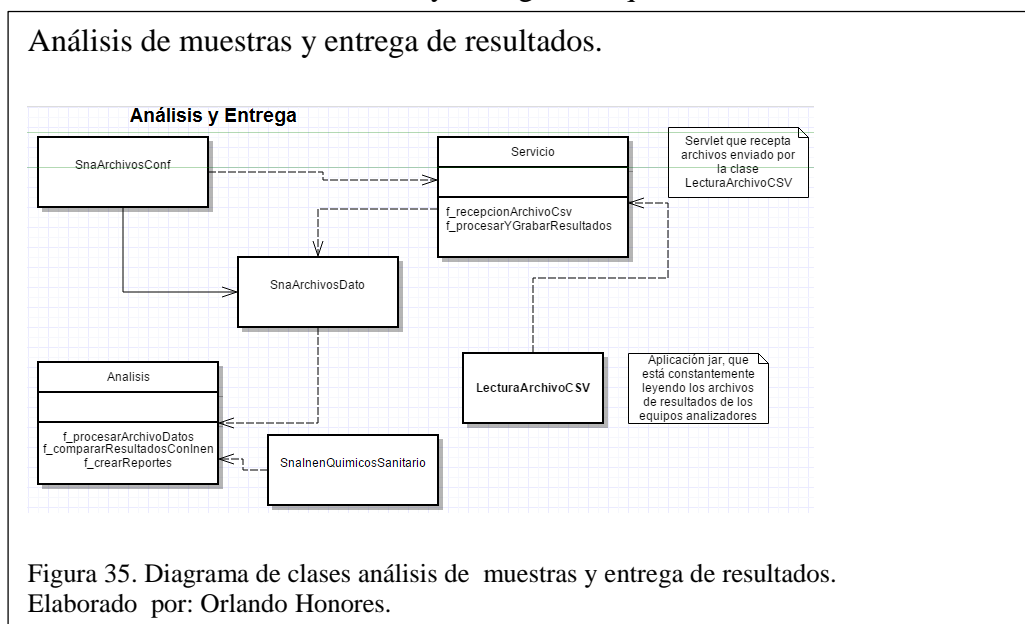


Diagrama de Clases Análisis de muestras y entrega de resultados.

La figura 35 representa en forma gráfica la relación entre las clases, que hacen referencia al módulo de Análisis y entrega de requerimientos



4.2 Creación de la aplicación usando Maven.

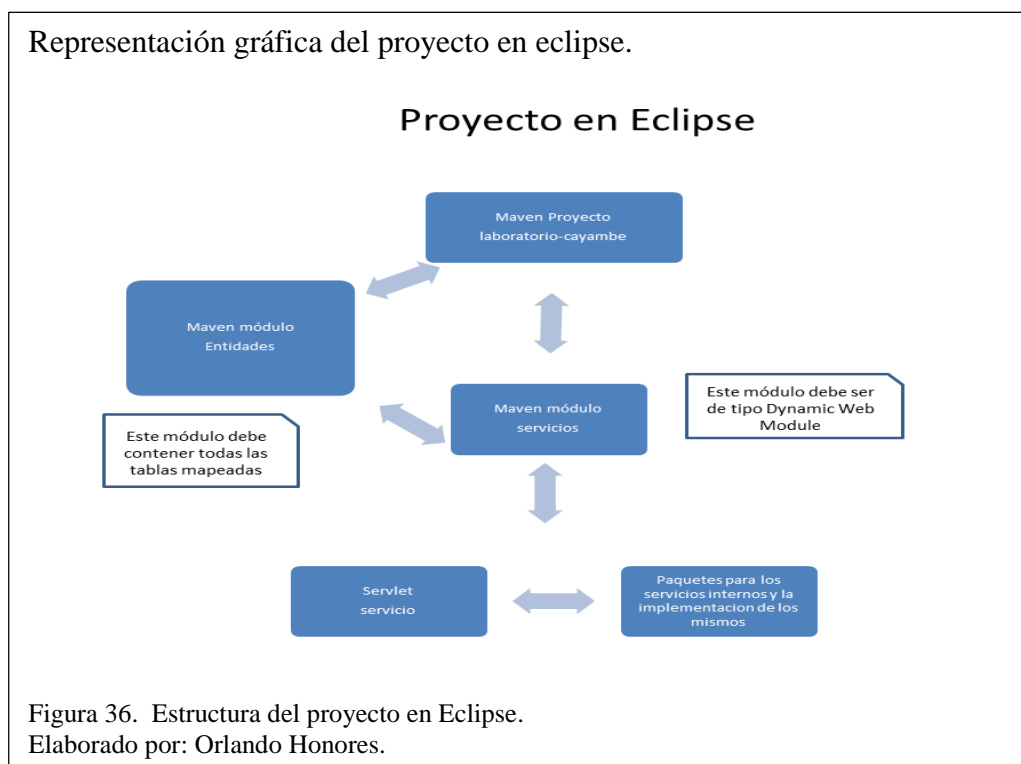
Para crear la aplicación java se utiliza un Proyecto Maven que contendrá las dependencias de todo el sistema informático. El mismo que estará acompañado de dos módulos en Maven, un módulo contendrá las entidades y relaciones mientras que el segundo estará conformado por los servicios que brindará el sistema, como el análisis de muestras de leche, recepción de archivos, validación de ingreso a la página principal entre otros.

Requisitos:

Eclipse versión 4.5,

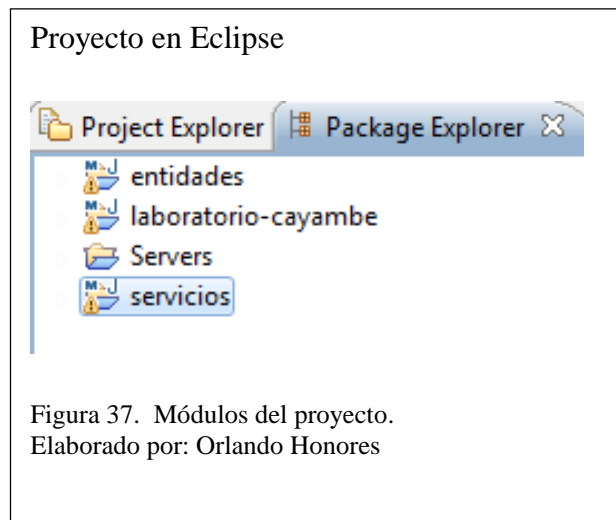
Apache Tomcat, versión 7.55; en el bin debe contener los script para que se inicie tanto en Linux como en Windows

La figura 36, muestra la estructura del proyecto, en esta imagen se aprecia la separación de las entidades de relación (módulo clientes) con la lógica del programa (módulo servicio). Para llegar a esta estructura se debe realizar lo siguiente:



Crear un proyecto maven llamado “Laboratio- cayambe”, luego se crea un módulo maven llamado “clientes”, indicando que el padre del proyecto es “Laboratorio- Cayambe”. Luego utilizando JPA tool, transforma las tablas a entidades de la base de datos y por último definimos el módulo maven llamado “servicios”, indicando que el padre del proyecto se llama “Laboratorio-Cayambe”.

En la figura 37, se muestra imagen del funcionamiento del software desarrollado sobre Eclipse.



Visualización de archivos de configuración:

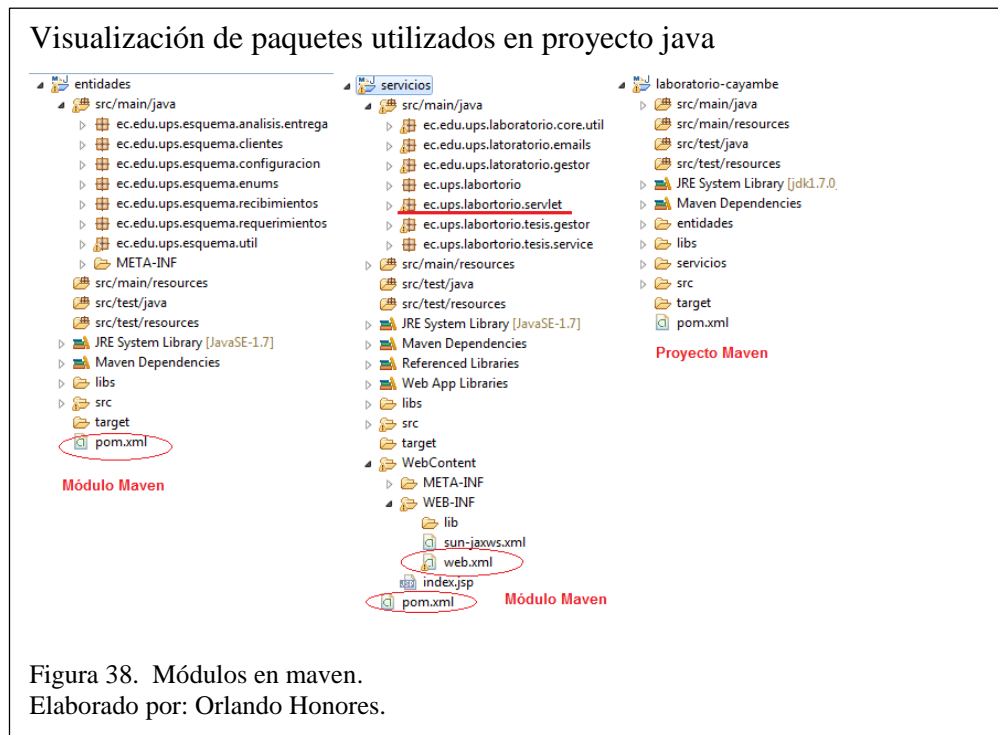
El descriptor de implementación: web.xml

- Las aplicaciones web Java utiliza un archivo de descriptor de implementación para determinar la forma en que las URL se asignan a los servlets y las direcciones URL que necesitan autenticación, entre otra información.
- Este archivo se llama web.xml y se encuentra en el WAR de la aplicación, concretamente, en el directorio WEB-INF/. web.xml forma parte del estándar de servlet para las aplicaciones web.

Project Object Model: pom.xml.

- Es un fichero XML, que es la “unidad” principal de un proyecto Maven.
- Contiene información acerca del proyecto, fuentes, test, dependencias, plugins, version, etc.

En la figura 38, se muestran los paquetes creados, los archivos de configuración pom.xml y web.xml



Entity Clases desde la Base de Datos.

En Eclipse, como en NetBeans permiten la creación de clases a partir de la base de datos, mediante una herramienta embebida llamada JPA, tools la cual permite escoger el origen de los datos, es decir las tablas e indicar el nombre del paquete en donde se almacenarán las entidades. El wizard que permite hacer esto, se lo obtiene haciendo click derecho en el proyecto y escoger JPA.

La figura 39, es una imagen que representa el tool de JPA, escogiendo “JPA Entities from Tablas”, para crear las entidades a clases java.

Creación de entidades en Eclipse

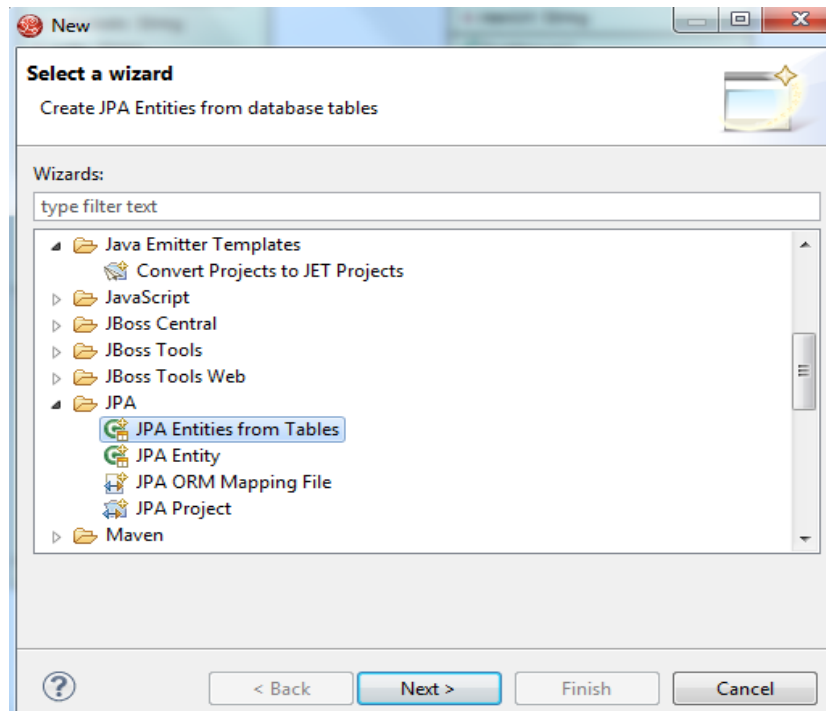


Figura 39. Muestra cómo se crea las entidades a partir de la herramienta JPA tolos.
Elaborado por: Orlando Honores.

Conexión con la base de datos.

A continuación se detallan los pasos para establecer una conexión con la base de datos:

Nos dirigimos a la carpeta conf, que se encuentra en el servidor Apache Tomcat, se localiza el archivo el siguiente archivo "context.xml" con el fin de modificarlo con el siguiente código:

```
<Resource name="jdbc/postgres" auth="Container"
type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://localhost:5432/ TESIS_DB_UPS_09092015"
username="postgres" password="root" maxActive="20"
maxIdle="10" maxWait="-1"/>
```

Este código crea un dataSource, el mismo que será llamado en el web.xml mediante el siguiente código, como se puede observar la referencia jdbc/postgres es la conexión con el dataSource en el archivo context.xml

```
<!-- JDBC DataSources (java:comp/env/jdbc) →  
  
<resource-ref>  
  
    <description>The default DS</description>  
  
    <res-ref-name>jdbc/postgres</res-ref-name>  
  
    <res-type>javax.sql.DataSource</res-type>  
  
    <res-auth>Container</res-auth>  
  
</resource-ref>
```

Instalación del servidor NodeJS,

El servidor NodeJS se lo instala con el siguiente comando en Linux

```
sudo apt-get install python-software-properties  
sudo add-apt-repository ppa:chris-lea/node.js  
sudo apt-get update  
sudo apt-get install nodejs npm
```

En Windows se debe descargar de la página oficial “NodeJs” el instalador para este sistema, un archivo con extensión "msi".

En la figura 40, se muestra la estructura de carpetas del servidor NodeJS

Estructura de carpetas del servidor NodeJS

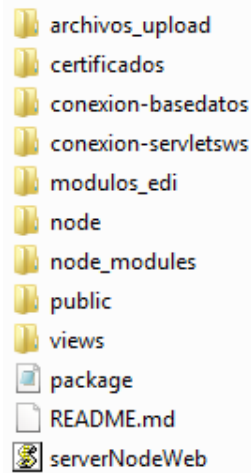


Figura 40 Estructura del servidor NodeJs.
Elaborado por: Orlando Honores.

Comandos para iniciar el servidor NodeJs.

En **Windows** se utiliza el siguiente comando para iniciarlo:

```
C:\ubicacion del servidor NodeJS> node serverNodeWeb.js
```

Para detener el servidor se presionan las teclas Ctrl+C

En **Linux** se utiliza el siguiente comando para iniciarlo:

```
\ubicacion del servidor nodejs# nohup node serverNodeWeb.js &
```

Para detener el servidor se hace un kill -9 pid, para obtener el pid se aplica el siguiente comando “ps -fe|grep node” del resultado que entregue, el primer número a la derecha es el pid.

Ejemplo: \ubicacion del servidor nodejs# kill -9 2323

Instalación del servidor Apache Tomcat.

El instalador se lo obtiene de la página oficial de Tomcat, según el sistema operativo se hace un download el instalador en formato zip.

Este archivo se lo descomprime en el directorio que se vaya a desarrollar la aplicación. Para el presente proyecto se lo descomprimió en la siguiente dirección:

- Linux
 - /home/laboratorio-leche/ apache-tomcat-7.0.55.zip
 - /home/laboratorio-leche/ apache-tomcat-7.0.55.zip
 - /home/laboratorio-leche/ unzip apache-tomcat-7.0.55
- Windows
 - C:/ laboratorio-leche/ apache-tomcat-7.0.55.zip, click derecho y descomprimir aquí.
 - C:/ laboratorio-leche/ apache-tomcat-7.0.55

En la carpeta apache-tomcat-7.0.55\webapps, se colocan los archivos con extensión war de la aplicación en java. Para obtener este archivo comprimido war se hace lo siguiente: Click derecho sobre el proyecto “servidores”, escoger exportar y se debe elegir el tipo de exportación WAR y la ubicación en donde se desee que se grabe el war.

Comandos para iniciar el servidor Tomcat.

En **Windows** se utiliza el siguiente comando para iniciarlo

- C:\ubicacion del servidor tomcat> startup.exe

Para detener el servidor

- C:\ubicacion del servidor tomcat> shutdown.exe

En **Linux** se utiliza el siguiente comando para iniciarlo:

- \ubicación del servidor tomcat\bin# ./startup.sh, previamente debe verificar que el archivo startup.sh tenga permisos de ejecución, si no lo tiene aplique el siguiente comando `chmod +x startup.sh`

Para detener el servidor:

- \ubicación del servidor tomcat\bin# ./shutdown.sh, previamente debe verificar que el archivo shutdown.sh tenga permisos de ejecución, si no lo tiene aplique el siguiente comando `chmod +x shutdown.sh`

Creación de Pantallas estándares para el ingreso, modificación, eliminación y visualización de registros

Partiendo del Diseño de Interfaz Abstracta, todas las páginas deben tener un mismo estilo para el ingreso de datos, modificación, anulación y visualización de resultados, se llegó a la conclusión de crear una herramienta propia que permita la creación de pantallas dinámicas según la tabla elegida y previamente configurada. Para la generación de esta herramienta se definió una tabla llamada `sna_configuracion`, que contendrá la información necesaria de la tabla como nombre del campo, nombre del label, que tipo de caja de texto aparecerá en el html, si es un campo de validación entre otros.

Para realizar esta pantalla, se valió de una librería llamada JQuery Mobile, que cuenta con gran cantidad de componentes que trabajan en forma rápida en html5 y la aplicación de un framework llamado AngularJS, que permite un control absoluto de la página web y cuenta con sus propias etiquetas.

4.3 Estándares de programación

A continuación se definen los nombres con que se identificarán los distintos objetos en el desarrollo de sistemas de información. Se toma de base nomenclatura Húngara.

Las descripciones que a continuación se detallan serán de carácter obligatorio durante todo el desarrollo del SGU.

Funciones

Los nombres de funciones irán precedidos por `F_` y siempre serán escritos en letras mayúsculas.

El nombre de la función debe ser descriptivo de acuerdo a la función que realice la misma.

Las funciones deben ser puras, es decir no debe acceder a la base de datos ni escribir variables de paquetes.

Después de la declaración de la función deberá comentar su utilización, el autor de la función, la fecha de creación, la fecha de la última modificación y el resultado que devuelve la función.

Al momento de realizar una actualización en la función se deberá respaldar la versión anterior especificando claramente la fecha de la actualización, además se deberá incluir la justificación de la actualización.

Los parámetros que se utilice la función irán precedidos por:

- pn_ Para los parámetros numéricos.
- pv_ Para los parámetros de caracteres.
- pf_ Para los parámetros de fecha.
- pb_ Para los parámetros boléanos.
- pt_ Para los parámetros type.
- pa_ Para los parámetros arreglo.

Ejemplo:

F_CALCULA_EDAD (pf_fecha_nacimiento in date)

Variables y Constantes

La siguiente disposición se aplica a las variables y constantes en todos los ámbitos de la programación.

Se especificará el alcance de la variable o la constante mediante:

- l Variables locales
- g Variables globales

Se especificará también el tipo de dato que almacenará la variable mediante la siguiente definición:

- n_ Para variable y constantes numéricas.
- v_ Para variables y constantes de caracteres.
- f_ Para variables y constantes de fecha.
- b_ Para variables y constantes boléanos.
- t_ Para variables y constantes type.
- a_ Para variables y constantes arreglo.

Además se comentará la utilización de la variable dentro del proceso.

Ejemplos:

Variable local numérica:

- ln_contador number; -- contador de procesos realizados
- Variable global booleana:
- gb_estado boolean; -- estado de autorización

Consultas

Las consultas (queries) se escribirán con todas las palabras reservadas en mayúsculas y los nombres de columna o constantes en minúsculas.

Las consultas deberán ser escritos identados.

Las consultas deberán ser comentadas indicando su utilización.

Ejemplo:

Selección de nombres y cédulas para el cálculo de la edad.

```
SELECT cedula, nombre
INTO lv_cedula, lv_nombre
FROM sistema.ad_persona
WHERE cedula = &pv_cedula
AND estado = 1
ORDER BY nombre
```

Formas y Reportes

Las formas y los reportes deberán ser comentados. Para los comentarios de las formas y reportes se utilizarán los respectivos campos que existen tanto a nivel de forma como a nivel de reporte. Dentro de estos comentarios deberá constar la siguiente información:

- Descripción de la forma.
- Autor.

- Fecha de creación.
- Fecha de modificación.
- Tablas.

Los nombres de las tablas de los sistemas se crearán con los siguientes prefijos:

Sna_ = Sistema Nacional Académico

Ejemplo:

SNA_PROYECTO_CAYAMBE_TESIS

Columnas de Tablas

Los nombres de las columnas de las tablas tendrán la siguiente estructura.

Si el nombre de la tabla tiene una sola palabra sin contar con el prefijo; los nombres de las columnas adoptarán como prefijo las tres primeras letras de la tabla.

Ejemplo:

Si la tabla se llama SNA_PROYECTO, las columnas deberán tener el prefijo PRO.

PRO_CODIGO, PRO DESCRIPCIÓN

Si el nombre de la tabla tiene dos palabras sin contar con el prefijo; los nombres de las columnas adoptarán como prefijo las dos primeras letras de la primera palabra y la primera letra de la segunda palabra.

Ejemplo:

Si la tabla se llama SNA_PROYECTO_ACADEMICO, las columnas deberán tener el prefijo PRA. PRA_CODIGO, PRA DESCRIPCIÓN

Si el nombre de la tabla tiene tres palabras sin contar con el prefijo, los nombres de las columnas adoptarán como prefijo la primera letra de cada palabra.

Ejemplo: Si la tabla se llama SNA_PROYECTO_ACADEMICO_NUEVO, las columnas deberán adoptar el prefijo PAN. PAN_CODIGO, PAN DESCRIPCIÓN.

Nombres de restricciones (constraints) en las tablas Clave primaria: Los constraints de clave primaria serán llamados con el nombre de la tabla seguidos del sufijo _PK.

Ejemplo:

Tabla SNA_PROYECTO_ACADEMICO

Constraint SNA_PROYECTO_ACADEMICO_PK

Si en nombre del constraint excede los 30 caracteres, el nombre de la restricción adoptara el prefijo de la aplicación más las tres primeras letras de las palabras que conformen el nombre de la tabla.

Ejemplo:

SNA_PRO_ACA_PK

Claves foráneas: Los constraints de clave foránea serán llamados con el nombre de la tabla más el nombre de la columna más el sufijo _FK.

Si la longitud del nombre del constraint excede los 30 caracteres, utilizar las tres primeras letras de cada una de las palabras que conforman el nombre de la tabla.

Si la Clave foránea de la tabla está compuesta por más de una columna, el nombre del constraint será el nombre de la tabla más el nombre de la primera columna que forma la clave foránea.

Ejemplo:

Tabla SNA_PROYECTO, columna FAC_CODIGO

Constraint SNA_PROYECTO_FAC_CODIGO_FK

Tabla SNA_PROYECTO_ACADEMICO_NUEVO, columna PAN_DESCRIPCIÓN

Constraint SNA_PRO_ACA_NUE_PAN_DESCRIPCION_FK

Tabla SNA_PROYECTO, columnas PRO_CODIGO, PRO_DESCRIPCIÓN

Constraint SNA_PROYECTO_PRO_CODIGO_FK

Campos no nulos: Los constraint de campos no nulos estarán formados por el nombre de la tabla más el nombre de la columna más el sufijo _NN.

Si la longitud del nombre del constraint excede los 30 caracteres, utilizar las tres primeras letras de cada una de las palabras que conforman el nombre de la tabla.

Ejemplo:

Tabla SNA_PROYECTO, columna PRO DESCRIPCIÓN

Constraint SNA_PROYECTO_PRO DESCRIPCIÓN_NN

4.4 Codificación del sistema

Autenticación de usuarios:

El sistema cuenta con una servlet llamado “servicio”, que valida la autenticación al sistema.

En esta clase hay una función llamada f_ `autenticarUsuario`, que valida el password ingresado por el usuario. Se usó un sistema de almacenamiento de bytes en una solo sentido para la contraseña, pero debido a que no es suficiente este mecanismo de seguridad se implementó una clave adicional generada en forma randómica. A este proceso se lo conoce como Salted Password Hashing que usa una función de hash y un dato variable denominado salt y es diferente para cada contraseña, en la base de datos se guarda el resultado de la función de hash junto con el salt, esto es, el resultado de SHA-512(contraseña+salt) y también el salt.

A continuación se muestra el parte del código donde realiza esta validación.

TesisG tesisG;

protected void f_autenticarUsuario(HttpServletRequest request,

HttpServletResponse response){

Map<String,Object> respuesta = new HashMap<String, Object>();

try{

String codigo =

request.getParameter(ObjectUtils.transformarDateToString(new
Date(), "yyyyddMM"));

```

String clave=
request.getParameter(ObjectUtils.transformarDateToString(new
Date(), "ddMMyyyy"));
Boolean condiciones =
Boolean.parseBoolean(request.getParameter("c"+ObjectUtils.transfor
marDateToString(new Date(), "MMyyyydd")));
if(!condiciones){
    respuesta.put("login", false);
    respuesta.put("mensaje", "Debe aceptar las condiciones");
    setJson(response,respuesta);
    return;
}
if(StringUtils.isBlank(codigo) && StringUtils.isBlank(clave)){
    respuesta.put("log in", false)
    respuesta.put("mensaje", "El usuario o password no deben
estar vacíos");
    setJson(response,respuesta);
    return;
}
/*****
 * Perfil usuario
 */
SnaPerfilusuario perfilusuario =
tesisG.obtenerDTOPorCriterios("perAlias", codigo,
SnaPerfilusuario.class);
    if(perfilusuario != null && perfilusuario.getPerEstado() !=
null && perfilusuario.getPerEstado().isACTIVO() &&
perfilusuario.getPerGenerator() != null &&
perfilusuario.getPerPassword() != null &&
CryptoUtils.authenticate(clave, perfilusuario.getPerPassword(),
perfilusuario.getPerGenerator()) &&
perfilusuario.getSnaVariosTipo().getVatEstado().isACTIVO() ){
        respuesta.put("login", true);

```

```

        respuesta.put("usuario", codigo );
respuesta.put("perfil",perfilusuario.getSnaVariosTipo().getVatId());
        setJson(response,respuesta);
        return;
    }else{
        respuesta.put("login", false);
        respuesta.put("mensaje", "Usuario o Password no existe, o el
        Usuario no está activo");
    }
}catch(Exception e){
    e.printStackTrace();
}
}

```

CAPÍTULO 5

IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA

5.1 Implementación del sistema informático

Una vez terminadas las etapas anteriores y después de haber concluido el sistema informático se procedió a darle un nombre “RIPLEY ZHOL UPS”, para identificarlo en esta etapa de implementación.

El Laboratorio de leche de Cayambe, designó una computadora con un sistema operativo Windows 7, con 4gigas en ram y con acceso a internet, con el objetivo de que se instale el sistema informático.

Lo primero que se instaló fue la base de datos Postgres versión 9.3, el nombre designado a la base de datos fue SNA_PROYECTO_CAYAMBE_TESIS. Seguido se procedió a instalar las herramientas utilizadas como NodeJS y el jdk de java 1.7

Una vez concluido con las herramientas descritas arriba, se definieron las estructuras de directorios:

Directorio de carpetas:

- C:/ sistema-laboratorio-leche/servidores/pagina-web, en esta ruta se encuentra la página web del sistema
- C:/ sistema-laboratorio-leche/servidores/servicios, en esta ruta se encuentran los servicios del sistema.
- C:/ sistema-laboratorio-leche/servidores/aplicación-jar, en esta ruta se aloja el middleware que se encarga de leer los archivos de texto y enviarlos al servicio.

Para un buen funcionamiento del sistema se instaló Chrome Versión 42 en las computadoras que no disponían de este browser, este navegador hace que la aplicación mejore su performance.

Una vez instalado el sistema el usuario puede acceder desde cualquier navegador, pero en este caso lo hace por medio del navegador Chrome.

5.2 Pruebas

Las tablas que se muestran a continuación, muestran casos de prueba de cada una de las funcionalidades del Portal Web, las correcciones realizadas se encuentran especificadas, el propósito de cada caso de prueba deberá coincidir con los resultados.

5.2.1 Pruebas funcionales.

Tabla 18.

Caso de Prueba: Inicio de sesión

Inicio de Sesión		
Propósito, comprobar ingreso de usuario a parte privada del Portal.		
Actor, cualquier usuario registrado (administrador, jefe de laboratorio, laboratorista)		
Datos de Prueba	Alias_usuario:	{ovhones}
	Contraseña :	{alien200525}
Pasos : <ul style="list-style-type: none">• Ingresar página principal del portal• Introducir nombre de usuario• Introducir contraseña• Hacer clic en el botón ingresar		
Resultado: el usuario registrado ingresa a la parte privada del PROYECTO, tiene su propio menú.		

Nota: Prueba realizada para iniciar sesión.

Elaborado por: Orlando Honores.

Tabla 19.

Caso de Prueba: Administrar usuarios

Administrar Usuarios		
Propósito, comprobar que el administrador pueda añadir o modificar Usuarios.		
Actor Administrador		
Datos de prueba		
Cedula:	{1714784251}	
Nombres:	{Orlando Honores}	

Email:	{ohonores@hotmail.com}
Dirección:	{Abelardo Andrade S1289}
Teléfono :	{3110367}
Nombre de usuario:	{ovhonores}
Contraseña:	{alien200525}
Re-Contraseña:	{alien200525}
Perfil:	{administrador}
Pasos <ol style="list-style-type: none"> 1. Ingresar página principal de la aplicación web. 2. Introducir nombre de usuario y contraseña 3. Hacer clic en el menú Mantenimiento->Usuarios->Laboratorista 4. En la pantalla mostrada hacer click en “Nuevo Registro” 5. Para modificar usuario, primero debe elegir el usuario de la lista mostrada. 6. Para cualquiera de las dos opciones ingresar los datos de prueba. 7. Presionar el botón guardar. 8. Presionar cerrar sesión 	
Resultado, el administrador puede añadir y modificar usuarios.	

Nota: Prueba realizada para administrar usuarios.

Elaborado por: Orlando Honores.

Tabla 20.
Gestionar acciones de perfil

Gestionar Acciones del Perfil
Propósito Comprobar que el administrador puede añadir, modificar o eliminar accesos y acciones de un perfil.
Actor: Administrador
Datos de Prueba Ingreso al sistema Nombre de Usuario: {ovhonores} Contraseña: {alien200525} Perfil: Administrador Acceso: Clientes->Productores. Acciones: todos
Pasos

1. Ingresar página principal del portal.
2. Introducir nombre de usuario y contraseña.
3. Hacer clic en el botón ingresar.
4. Escoger el menú Mantenimiento->Varios Tipos.
5. Desplegar “Grupos de Perfiles” y seleccionar el perfil Administrador.
6. Clic en Administración de Menús y permisos
7. En la nueva pantalla escoger Clientes->Productor de la lista “Menús del Sistema”
8. Clic en Preview Menú, verificar que el menú Clientes solo contenga el submenú Productor.
8. Dirigirse a Permisos Establecidos y Productor.
11. Luego en el bloque “Permisos”, escoger todos. (o sea seleccionar todos) y click en Agregar
13. Presionar Actualizar “Menú para el perfil Administrador”
14. Salir del sistema.

Resultado El administrador puede añadir, modificar o eliminar atributos extras de cualquier perfil seleccionado.

Nota: Prueba realizada para administrar el perfil de usuarios.
Elaborado por: Orlando Honores.

La figura 41, es una imagen de la página de inicio del sistema informático, en donde se puede visualizar que contiene una pantalla de ingreso de usuario y password.



La figura 42, es una imagen de la página de configuración de tablas del sistema.

Pantalla de configuración de páginas dinámicas

Figura 42. Configuración de pantallas dinámicas.
Elaborado por: Orlando Honores.

El ingreso de datos en el sistema informático cuentan con una pantalla estándar con el mismo formato, ver figura 43.

Ingreso de datos

Figura 43. Pantalla ingreso de datos.
Elaborador por: Orlando Honores.

CONFIGURACIÓN | VARIOS TIPOS

Buscar Cliente/Empresa/Proveedor LABORATORIO

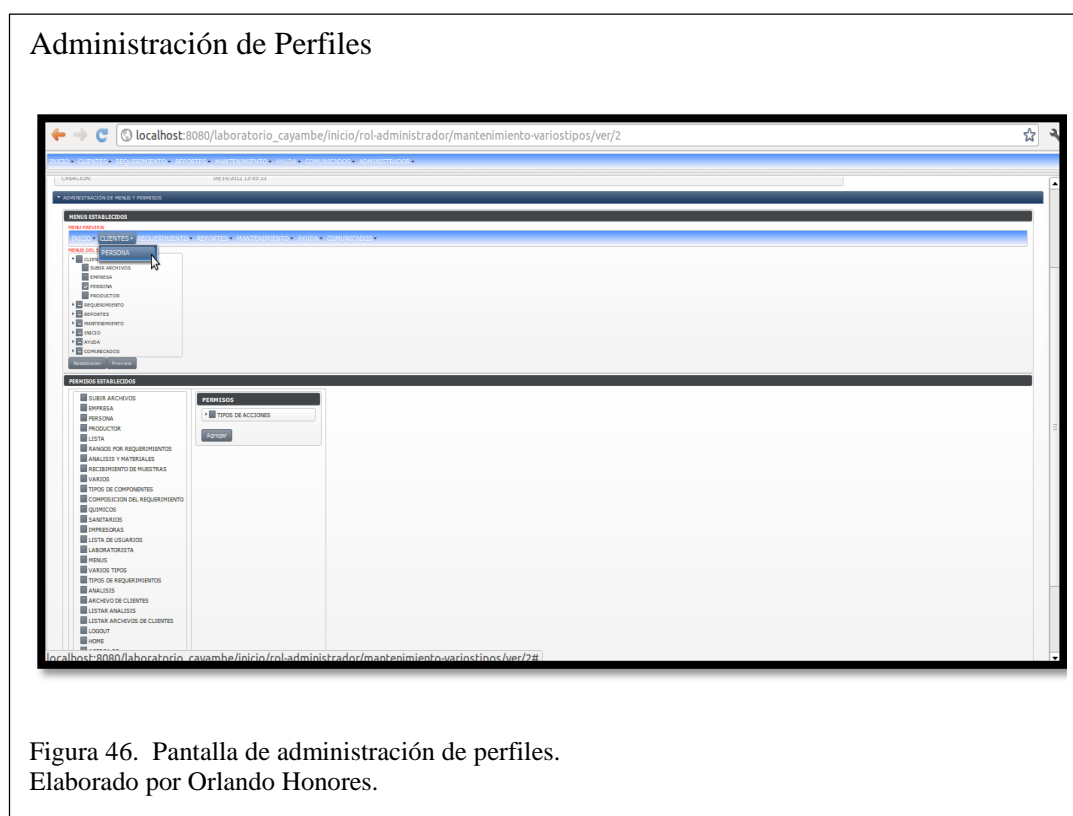
Ingreso de datos Listar datos

Referencia	ID	ESTADO	NOMBRE	CREACIÓN	MODIFICACIÓN	Modificar
ACCIONES	2	I	ACCIONES	2015-06-19T22:30:50.251Z	2015-06-27T13:54:11.383Z	<input type="button" value="M"/>
ACCIONES >> ELIMINACIÓN DE LA BASE	5	A	ELIMINACIÓN DE LA BASE	2015-06-19T23:03:36.796Z	2015-06-19T23:03:36.796Z	<input type="button" value="M"/>
ACCIONES >> INGRESO	3	A	INGRESO	2015-06-19T22:57:57.073Z	2015-06-19T22:57:57.073Z	<input type="button" value="M"/>
ACCIONES >> MODIFICACION	4	A	MODIFICACION	2015-06-19T23:03:04.451Z	2015-06-19T23:03:04.451Z	<input type="button" value="M"/>
GRUPO DE PERFILES	10	A	GRUPO DE PERFILES	2015-06-20T03:55:47.357Z	2015-06-25T21:08:15.058Z	<input type="button" value="M"/>
GRUPO DE PERFILES >> ADMINISTRADOR	11	A	ADMINISTRADOR	2015-06-20T03:57:02.605Z	2015-06-20T03:57:02.605Z	<input type="button" value="M"/>
GRUPO DE PERFILES >> CLIENTE	13	A	CLIENTE	2015-06-20T18:57:22.683Z	2015-06-21T04:31:48.833Z	<input type="button" value="M"/>
GRUPO DE PERFILES >> CLIENTE >> EMPRESA	49	A	EMPRESA	2015-06-27T22:44:33.037Z		<input type="button" value="M"/>
GRUPO DE PERFILES >> CLIENTE >> PRODUCTOR	50	A	PRODUCTOR	2015-06-27T22:45:11.494Z		<input type="button" value="M"/>
GRUPO DE PERFILES >> USUARIO	12	A	USUARIO	2015-06-20T03:57:13.104Z	2015-06-25T21:54:06.827Z	<input type="button" value="M"/>
ts+8lldecdsdd	39	A	ts+8lldecdsdd	2015-06-21T19:30:26.678Z	2015-06-27T13:53:15.676Z	<input type="button" value="M"/>
TIPOS DE IDENTIFICACIÓN	7	A	TIPOS DE IDENTIFICACIÓN	2015-06-20T03:51:32.567Z	2015-06-20T03:51:32.567Z	<input type="button" value="M"/>
TIPOS DE IDENTIFICACIÓN >> CÉDULA	8	A	CÉDULA	2015-06-20T03:52:29.713Z	2015-06-20T03:52:29.713Z	<input type="button" value="M"/>
TIPOS DE IDENTIFICACIÓN >> RUC	9	A	RUC	2015-06-20T03:52:45.148Z	2015-06-25T17:41:06.471Z	<input type="button" value="M"/>

Figura 45. Pantalla de informe.

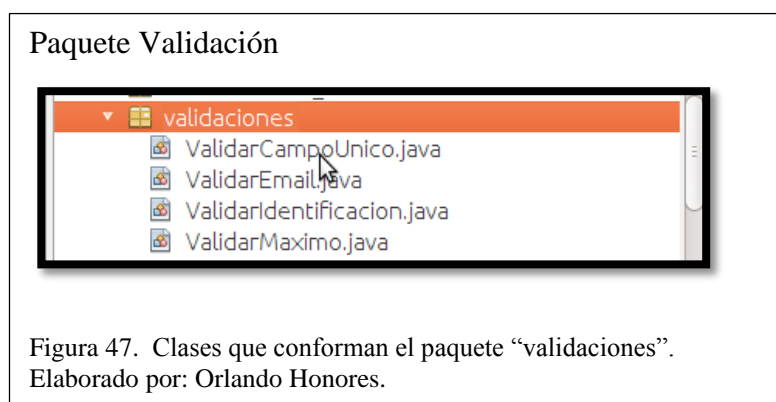
Elaborado por: Orlando Honores.

La Figura 46, describe los resultados de análisis de leche por perfiles



5.2.2 Pruebas de validación.

Para estas pruebas se creó un paquete llamado “validaciones”, en la figura 47 se contemplan las clases creadas para la validación durante o al final del desarrollo para determinar si satisface los requisitos iniciales.



Las siguientes líneas de códigos ubicadas al comienzo de la clase permiten que estas clases sean llamadas desde el front-end de la aplicación.

- @FacesValidator("org.ups.ValidarCampoUnico")
- @FacesValidator("org.ups.ValidarEmail")
- @FacesValidator("org.ups.ValidarIdentificacion")
- @FacesValidator("org.ups.ValidarMaximo")

La figura 48, representa la validación de la cédula cuando un cliente la ingresa incorrectamente.

Validación de Cédula

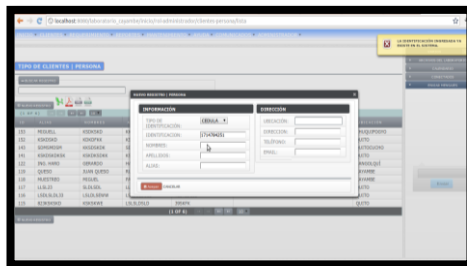


Figura 48. Pantalla de validación de la cédula.
Elaborado por: Orlando Honores.

La figura 49, muestra un mensaje de validación cuando el email fue incorrecto

Validación de Email

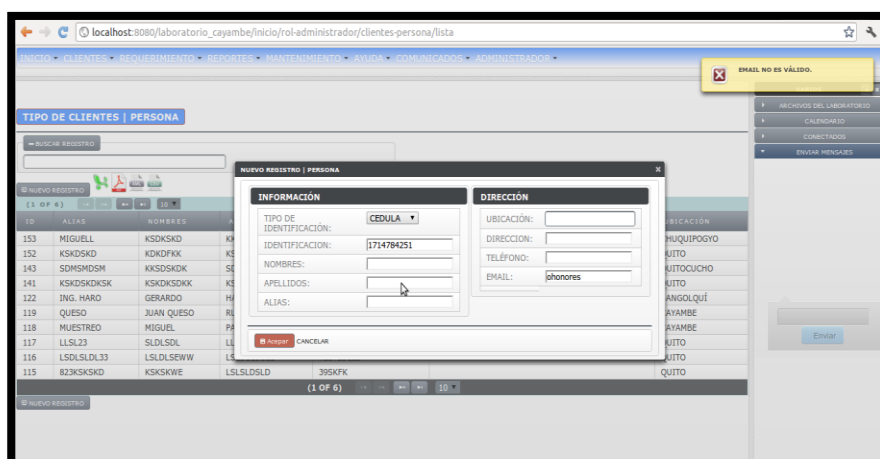


Figura 49. Pantalla ingreso de email y validación del mismo.
Elaborado por: Orlando Honores.

CONCLUSIONES

El desarrollo del sistema informático propuesto en el presente proyecto AngularJs, JSF, Postgres, Tomcat, NodeJS, Hibernet, permitieron resolver con mayor facilidad y rapidez validaciones, conversiones, mensajes de error internacionalización y el uso de un modelo de eventos en el lado del servidor, por su robustez y por ser software libre.

Para lograr desarrollar el sistema informático propuesto, ajustado a los requerimientos funcionales del usuario, se partió de un buen análisis de requerimientos, que involucró de manera directa a todos los actores miembros del Laboratorio de Leche de Cayambe de la Universidad Politécnica Salesiana. Esto permitió captar mediante técnicas de recopilación de datos como: la encuesta, revisión de documentos y asistencia a charlas de capacitación, todas las necesidades en cuanto a mejora y priorización de la ejecución de los procesos, los mismos que se convirtieron en las funcionalidades del sistema informático Web implementado.

Para la construcción del sistema informático, se aplicaron modelos y diagramas propuestos en la metodología OOHDM, por ser simples y claros, lo que permitió reflejar toda la lógica funcional del nuevo sistema informático implementado.

Con la creación del sistema de administración de contenidos, menús, accesos y configuración de archivos, se logró que el ingreso, actualización y consulta de datos sea más fácil, rápido y eficaz, por parte de los laboratoristas autorizados.

La implementación de políticas de seguridad como es el uso de encriptación MD5 para las contraseñas de usuarios autorizados, permitió obtener un alto nivel de seguridad, confiabilidad e integridad en el manejo de la información que gestiona el Laboratorio de Leche.

Se utilizó ciertas funciones dentro Postgres gracias a su gran escalabilidad, haciéndolo idóneo para su uso en sitios web, lo que ayudó a no saturar al servidor de base de datos con la ejecución de consultas en formas directas, esto ha permitido ahorro de tiempo en la implementación de la aplicación y que tenga un mayor rendimiento.

Para la implementación del servicio de chat, se utilizó un framework que combina JavaScript Socket.IO y NodeJS tecnología que ha permitido que cuando el servidor recibe un mensaje, de uno de los usuarios de chat, sea necesario enviar dicho mensaje a los otros usuarios sin necesidad de que lo hayan solicitado. Con la técnica de programación “Comet”, el servidor puede enviar mensajes a los usuarios a medida que se publican en lugar de esperar a los usuarios para sondear el servidor de mensajes nuevos.

RECOMENDACIONES

Es recomendable, si el sistema es gestionado para que funcione sobre internet, adquirir un certificado de seguridad para garantizar que la información viaje de forma segura a través de la web, a fin de evitar infiltraciones no autorizadas sobre la misma.

Se recomienda elaborar una normativa interna sobre el uso ético de los datos registrados y sobre la información generada a través del nuevo sistema informático implementado, por considerar información confidencial y delicada que solo es de interés para el cliente.

Como parte complementaria al sistema informático desarrollado, se recomienda complementarlo, como una segunda etapa, la construcción del módulo de facturación y de análisis de proyecciones sobre costos y manejo de clientes.

LISTA DE REFERENCIAS

- Diagrama de secuencia. (23 de junio de 2015). Recuperado el 13 de julio de 2015, de Wikipedia, La enciclopedia libre: https://es.wikipedia.org/w/index.php?title=Diagrama_de_secuencia&oldid=83359429
- Eclipse (software). (30 de junio de 2015). Recuperado el 13 de julio de 2015, de Wikipedia, La enciclopedia libre: [https://es.wikipedia.org/w/index.php?title=Eclipse_\(software\)&oldid=83512635](https://es.wikipedia.org/w/index.php?title=Eclipse_(software)&oldid=83512635)
- JasperReports. (15 de mayo de 2015). Recuperado el 13 de julio de 2015, de Wikipedia, La enciclopedia libre: <https://es.wikipedia.org/w/index.php?title=JasperReports&oldid=82437001>
- Java (lenguaje de programación). (2 de junio de 2015). Recuperado el 14 de julio de 2015, de Wikipedia, La enciclopedia libre: [https://es.wikipedia.org/w/index.php?title=Java_\(lenguaje_de_programaci%C3%B3n\)&oldid=83353657](https://es.wikipedia.org/w/index.php?title=Java_(lenguaje_de_programaci%C3%B3n)&oldid=83353657)
- Java EE. (6 de abril de 2015). Recuperado el 13 de julio de 2015, de Wikipedia, La enciclopedia libre: https://es.wikipedia.org/w/index.php?title=Java_EE&oldid=81225240
- Microsoft Windows. (10 de julio de 2015). Recuperado el 13 de julio de 2015, de Wikipedia, La enciclopedia libre: https://es.wikipedia.org/w/index.php?title=Microsoft_Windows&oldid=83693008
- Modelo entidad-relación. (27 de junio de 2015). Recuperado el 13 de julio de 2015, de Wikipedia, La enciclopedia libre: https://es.wikipedia.org/w/index.php?title=Modelo_entidad-relaci%C3%B3n&oldid=83453325

- PostgreSQL. (13 de junio de 2015). Recuperado el 13 de julio de 2015, de Wikipedia, La enciclopedia libre: <https://es.wikipedia.org/w/index.php?title=PostgreSQL&oldid=83157648>
- Tomcat. (8 de julio de 2015). Recuperado el 13 de julio de 2015, de Wikipedia, La enciclopedia libre: <https://es.wikipedia.org/w/index.php?title=Tomcat&oldid=83652784>
- Ubuntu. (30 de junio de 2015). Recuperado el 13 de julio de 2015, de Wikipedia, La enciclopedia libre: <https://es.wikipedia.org/w/index.php?title=Ubuntu&oldid=83507397>
- Artech Consultores S.R.L. (2013). Java (F) - Pool de conexiones. Recuperado el 13 de julio de 2015, de GXtechnical Home: <http://library.gxtechnical.com/gxdisp/pub/genexus/java/docum/manuals/8.0/mjavaf5.htm>
- Ávila Cesar, L. (2001). MODELANDO CON UML Principios y Aplicaciones. Trujillo-Perú: RJ S.R Ltda.
- Berzal, F. (2014). Base de Datos ||. Recuperado el 13 de julio de 2015, de Fernando Berzal: <http://elvex.ugr.es/idbis/db/docs/intro/C%20Modelado%20de%20datos.pdf>
- Carneiro, L., Coffin, M., Coewan, D., & Lucena, C. (1994). ADVCharts: a Visual Formalism for Highly Interactive Systems.
- Code School. (2015). AngularJS — Superheroic JavaScript MVW Framework. Recuperado el 13 de julio de 2015, de Shaping up with Angular.js: <http://campus.codeschool.com/courses/shaping-up-with-angular-js/intro>
- Cowan, D. D., & P.Lucena, C. (1995). Abstract Data Views, An Interface Specification Concept to Enhance Design for Reuse.
- Craig, L. (2008). UML y Patrones Introducción al análisis y diseño orientado a objetos.

Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante. (26 de 06 de 2014). El ciclo de vida de JSF. Recuperado el 13 de 07 de 2015, de <http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion03-apuntes.html>

expresions, P. r. (2010). The Premier website about Regular Expressions. Recuperado el 2014, de <http://www.regular-expressions.info/>

Foss. (s.f.). Soluciones analíticas para análisis y control de calidad alimentaria - FOSS. Recuperado el 2015, de <http://www.foss.es/industry-solution/central-milk-testing/products>

Foss-bactoscan. (1956). Individual bacteria count raw milk analysis with BactoScan bacteria analyser. Recuperado el 2015, de <http://www.foss.dk/industry-solution/products/bactoscan-fc/>

Foss-fossomatic. (s.f.). Fossomatic™ FC - the best in somatic cell counting. Recuperado el 2015, de <http://www.foss.dk/industry-solution/products/fossomatic-fc/>

foss-milkscan. (s.f.). MilkoScan™ FT2 advanced infrared milk analyser for profit in dairy production. Recuperado el 2015, de <http://www.foss.dk/industry-solution/products/milkoscan-ft2/>

Gamma, Helm, R., & Vlissides, R. J. (1995). Design Patterns: Elements of reusable object-oriented software. Addison Wesley.

GmbH, v. (26 de 06 de 2011). JSF (JavaServer Faces) - Tutorial. Recuperado el 13 de 07 de 2015, de <http://www.vogella.com/tutorials/JavaServerFaces/article.html>

GmbH, v. (07 de marzo de 2015). JPA 2.0 with EclipseLink - Tutorial. Recuperado el 13 de julio de 2015, de <http://www.vogella.com/tutorials/JavaPersistenceAPI/article.html>

hispalinux.e. (2015). ¿Qué es el Software Libre? | Hispalinux. Recuperado el 13 de julio de 2015, de <http://hispalinux.es/SoftwareLibre>

- INEN. (2015). Servicio Ecuatoriano de Normalización | Ecuador. Recuperado el 13 de julio de 2015, de <http://www.normalizacion.gob.ec/wp-content/uploads/downloads/2013/11/obligatorias2013.pdf>
- Lapiente, M. J. (2013). Modelo OOHDM. Recuperado el 2015, de <http://www.hipertexto.info/documentos/oohdm.htm>
- Node.js Foundation. (2015). Node.js. Recuperado el 13 de 07 de 2015, de <https://nodejs.org/>
- Object Management Group, Inc. (22 de 05 de 2015). Unified Modeling Language (UML). Obtenido de <http://www.uml.org/>
- Ocpsoft. (2010). Is your web application secure? HTTP attacks are real, and dangerous. Recuperado el 2014, de <http://ocpsoft.org/opensource/secure-your-applications-url-based-attacks-are-real-and-dangerous/>
- ocpsoft. (2010). OCPsoft & Persist and pass FacesMessages over multiple page redirects Comments Feed. Recuperado el 2014, de <http://ocpsoft.org/java/persist-and-pass-facesmessages-over-page-redirects>
- Ocpsoft. (2010). Serving dynamic file content with PrettyFaces. Recuperado el 2014, de <http://ocpsoft.org/prettyfaces/serving-dynamic-file-content-with-prettyfaces>
- Regemim. (2009). java - Elegantly handling constraint violations in EJB/JPA environment. Recuperado el 2015, de <http://stackoverflow.com/questions/2519902/elegantly-handling-constraint-violations-in-ejb-jpa-environment>
- Scwabe, D. (2008). The Object-Oriented Hypermedia Design Model (OOHDM). Recuperado el 2015, de <http://www.oohdm.telemidia.puc-rio.br/>
- Slideshare. (2010). PrettyFaces: SEO, Dynamic, Parameters, Bookmarks, Navigation for JS. Recuperado el 2014, de <http://www.slideshare.net/lincolnthree/prettyfaces-seo-dynamic-parameters-bookmarks-and-navigation-for-jsf-jsf2-urlrewrite>

socket.io. (2015). Socket.IO. Recuperado el 13 de julio de 2015, de <http://socket.io/>

TIBCO Software, Inc. (2015). iReport Designer, Jaspersoft Community. Recuperado el 13 de julio de 2015, de <http://community.jaspersoft.com/project/ireport-designer>

Vogella. (2010). Java Regex Tutorial. Recuperado el 2014, de <http://www.vogella.com/articles/JavaRegularExpressions/article.html>

ANEXOS

Anexo 1. Manual de usuarios

Introducción

El presente manual de Usuario, tiene como finalidad dar a conocer de una manera detallada y sencilla la estructura de la aplicación Web del Laboratorio de Leche de Cayambe para que cualquier los usuarios del Laboratorio de Cayambe puedan el sistema, el sitio fue diseñado para que el usuario pueda, de una forma intuitiva y sin mayor capacitación, realizar búsquedas eficientes dentro del mismo.

Objetivos

Brindar una descripción clara y detallada sobre el funcionamiento y uso de los distintos elementos de la Página Web del Laboratorio de Leche de Cayambe de la Universidad Politécnica Salesiana al usuario en los diferentes tipos de búsqueda dentro de la Pagina Web, para así poder apoyarlo en la búsqueda de información

Requerimientos de Software:

El sitio está diseñado para los siguientes navegadores:

- a) Internet Explorer 7 o superior
- b) Mozilla Firefox 3 o superior
- c) Google Chrome 21 o superior

La resolución de la pantalla debe estar como mínimo en 768 x 1024

Página principal.

La siguiente página es la que se muestra cuando ingresa al sistema.

Esta página obliga al usuario a ingresar clave y contraseña. Solo usuarios que tengan al sistema pueden acceder a los recursos del mismo.

Página principal



Elaborado por: Orlando Honores

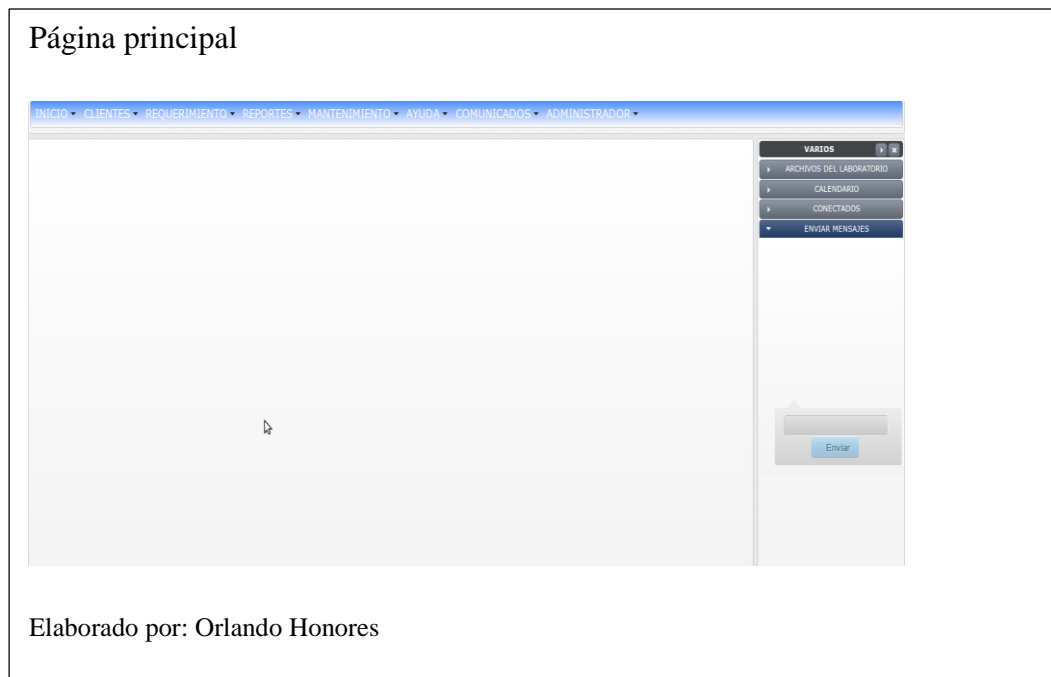
Acceso

En la opción de acceso dependiendo del tipo de usuario este debe introducir su nombre de usuario y contraseña para luego presionar el botón Ingresar.

Acceso

Elaborado por: Orlando Honores

Visualizará la siguiente pantalla, el menú que se observa varía según los permisos concedidos por el Administrador.



Menú Principal

La aplicación Web cuenta con el siguiente menú:

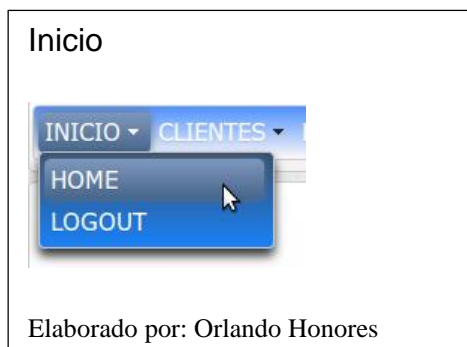


Inicio

Este menú está formado por dos submenús:

Home.

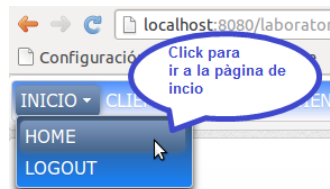
Indica que estamos en la raíz del proyecto.



Logout

Permite cerrar la sesión, y nos re direcciona a la página /login/

Menús



Elaborado por: Orlando Honores

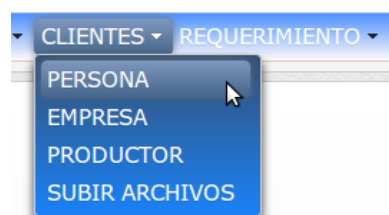
Clientes

Está formado por Persona, Empresa y Productor. Permitiendo ver, ingresar, editar y dar de baja registros.

Persona

Este submenú nos muestra todos los clientes de tipo persona.

Menú



Elaborado por: Orlando Honores

Menú persona



Elaborado por: Orlando Honores

Persona

ID	ALIAS	NOMBRES	APELLIDOS	IDENTIFICACION	DIRECCION	UBICACION
167	ALIAS	ORLANDO	HONORES	LICENCIA_167		CHILLOGALLO
166	HININNII	ORLANDO	HININNII	LICENCIA_166		CHILLOGALLO
161	JESUS_	JESUS	AGUIRE	CODIGO_161		CHILLOGALLO
160	MARITA	MARTITA	CEVALLOS	CODIGO_160		QUITO
159	SDDSDS	MIGUEL	PAEZ	CODIGO_159		QUITO
158	PKO	MIGUEL	PAEZ	CODIGO_158		QUITO
157	PMA	MIGUEL	PAEZ	CODIGO_157		QUITO
153	MIGUELL	KSDKSKD	KKSDKS	CODIGO_153		CHUQUIPOGOYO
152	KSKDSKD	KDKDFKK	KSDKKK			QUITO
143	SDMSMDSM	KKSDSKDK	SDSKDK	JDKSKDKS		QUITOCUCHO

Elaborado por: Orlando Honores

Características de la página estándar para visualizar los registros, ingreso, modificación y eliminación de un registro.

Características de la página

ID	ALIAS	NOMBRES	APELLIDOS	IDENTIFICACION	DIRECCION	UBICACION
167	ALIAS	ORLANDO	HONORES	LICENCIA_167		CHILLOGALLO
166	HININNII	ORLANDO	HININNII	LICENCIA_166		CHILLOGALLO
161	JESUS_	JESUS	AGUIRE	CODIGO_161		CHILLOGALLO
160	MARITA	MARTITA	CEVALLOS	CODIGO_160		QUITO
159	SDDSDS	MIGUEL	PAEZ	CODIGO_159		QUITO
158	PKO	MIGUEL	PAEZ	CODIGO_158		QUITO
157	PMA	MIGUEL	PAEZ	CODIGO_157		QUITO
153	MIGUELL	KSDKSKD	KKSDKS	CODIGO_153		CHUQUIPOGOYO
152	KSKDSKD	KDKDFKK	KSDKKK			QUITO
143	SDMSMDSM	KKSDSKDK	SDSKDK	JDKSKDKS		QUITOCUCHO

Elaborado por: Orlando Honores

Nuevo registro

Existe un botón llamado “Nuevo Registro”, que permite el ingreso de registros al sistema. El diálogo es mostrado una vez que el botón “Nuevo Registro” es presionado.

Nuevo registro

Al dar click en Nuevo Registro, se muestra un Diálogo

Graba el registro

Cancelar el ingreso de datos

Elaborado por: Orlando Honores

Buscar registro

Es un campo de texto autocomplete, que funciona con el ingreso de al menos 3 caracteres.

Buscar registros

El campo de búsqueda funciona ingresando al menos 3 caracteres.

TIPO DE CLIENTES PERSONA
3 - 1714784251 - TERRYS - ORLANDO - HONORESD
166 - LICENCIA_166 - HININNII - ORLANDO - HINIINNII
167 - LICENCIA_167 - ALIASS - ORLANDO - HONORES

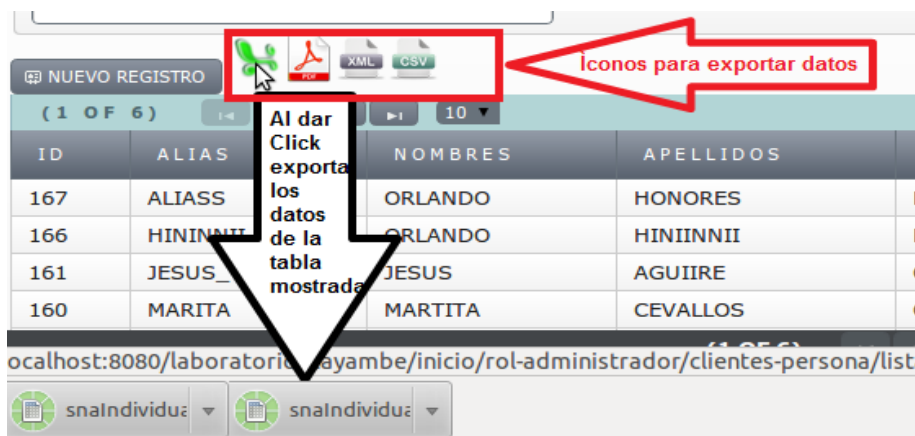
APELLIDOS	IDENT
HONORES	LICENC
HINIINNII	LICENC

Elaborado por: Orlando Honores

Exportar datos.

Los íconos que se encuentran en la parte superior de la tabla permiten exportar datos.

Exportar Datos

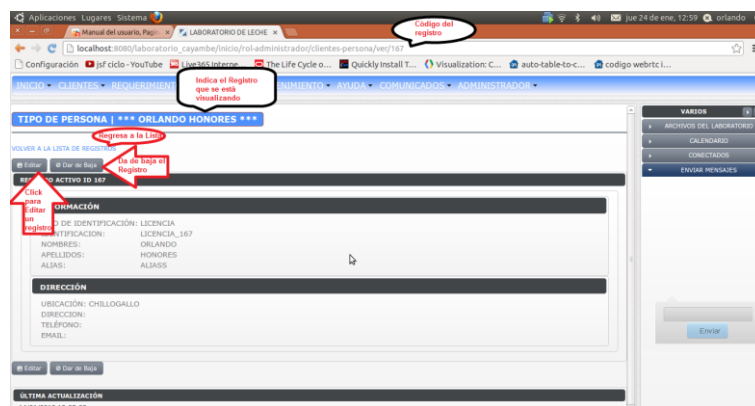


Elaborado por: Orlando Honores

Ver registros

Par ver el detalle de un registro, hacer click en el registro que desea visualizar.

Ver registros



Elaborado por: Orlando Honores